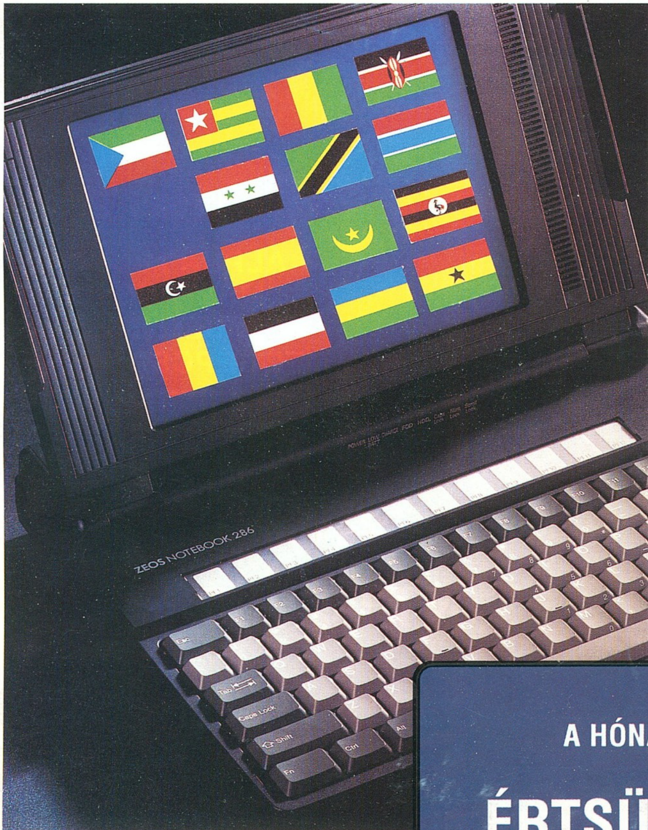
1991 / ÁPRILIS

ÁRA: 196 FT

ALAPLAP

MIKROSZÁMÍTÓGÉP MAGAZIN MÁGNESLEMEZ MELLÉKLETTEL



Vissza a kézíráshoz

Hangok a háttérben

A problémától a programig

Wampum varázslat?

Úton a pontos címhez

A pascalosok réme

A HÓNAP TÉMÁJA:

ÉRTSÜNK SZÓT!

A MÁGNESLEMEZEN:

Háttérzene-bona
Címtárkezelő Kisokos
PopDrop memóriamenedzser
Clipper Decompiler
ASCII konverziók

Újdonságlesen a Cebiten

Magyarul beszél a Progress

Shareware-országi utazások

ALPHA MICROSYSTEMS
AMERIKAI CSÚCSTECHNOLÓGIA
MAGYARORSZÁGON



CREATIVE COMPUTER SOLUTIONS

ALAPLAP

Mikroszámítógép magazin mágneslemez melléklettel

Megjelenik havonta

Főszerkesztő: Faklen Pál

Szerkesztő: Varga János

Olvasószerkesztő: Jakab Ágnes

Főmunkatárs: Kis János

A mágneslemez melléklet és a Közkincs rovat szerkesztője: Vékony Tamás

A szerkesztőbizottság tagjai:
Barna László
Boros György
Broczkó Péter
Brüll Károly
Farkas Ernő
Herczeg József
Horváth Imre
Kassay Árpád
Kovács P. Attila
Kónya László
Pintér Gábor
Zoltai Péter

Szerkesztőség, kiadó és hirdetésszervezés: XIV., Erzsébet királyné útja 17. Budapest 1251 Telefon/Fax: 252-1733

Felelős kiadó: Sebestyén Ilona igazgató Cédrus Informatikai Rt.

Nyomdai előkészítés: Tipoprint Kft., Budapest
Nyomtatás: Zalai Nyomda, Zalaegerszeg
Felelős vezető: Galla József

Terjeszti a Magyar Posta. Előfizethető a hírlapkézbesítő postahivataloknál és a Posta Hírlapelőfizetési és Lapellátási Irodájánál (XIII., Lehel u. 10/a, Budapest 1900), vagy átutalással a 215-96162 pénzforgalmi számra.
Példányonkénti ár: 196,- Ft
Évi előfizetési díj: 2352,- Ft

Külföldre terjeszti a Kultúra, Pf. 149, Budapest 1389

HU ISSN 0865-9788

A HÓNAP TÉMÁJA: ÉRTSÜNK SZÓT!

EGÉRFOGÓ

SZOFTVERTÉKA

KÖZKINCS

SOLARSOFT LEMEZKALAUZ

GÉPRAJZ

VÍRUSŐRJÁRAT

KILÁTÓ

KIRAKAT

ALAPJÁRAT

PROGRAMOZÁSTECHNIKA

MÁGNESLEMEZ MELLÉKLET



Elnézést azoktól a tisztelt olvasóktól, akik az Alaplap szellemétől meglehetősen idegennek érzik az irodalmi párhuzamokat. Engedtessék meg mégis sokak emlékébe idéznem Karinthy Frigyesnek a fenti három versszakot logikai sorba fűző, örökbecsű „Műfordítás" c. remekét. A mindössze két oldalnyi terjedelmű oknyomozó írás maga az információelmélet, annak példázata, miként is ért szót egymással két ember, majd másik kettő, megint kettő és így tovább: miként lesz — némi német közvetítéssel — az Ady-Gangeszből Herz-szalámi.

Jöttem a Gangesz partjairól,
Hol álmodoztam déli verőn,
A szívem egy nagy harangvirág
S finom remegések: az erőm.

Ufer, a zsidó kupléíró
Aludt a folyosón mélyen
Barátja, Herz, biztatta
Hogy ne remegjen, ne féljen.

A Herz-féle szalámiban
Sokkal sűrűbb a só,
Mint más hasonló terményekben
Hidd el, ó, nyájas olvasó!

Természetesen nem azt akarom mondani, hogy a Karinthy-féle parabola a számítógépes fordítás során is teljes mértékben igazolódik, de vitathatatlan, hogy az automatizált, nagy rendszerek részleteiben mindenkor ott rejtőzik a kisördög — maga az ember. Talán ettől is érdekes ez az egész, ettől emberi.

Vajon miért kell a „felhasználóbarát" programok mellé vaskos kézikönyveket applikálni? Mert az információt használható, futtatható formába önteni könnyebb, mint mindezt megértetni egy másik emberrel. Egy menüpontot definiálni rutinfeladat a programozó számára. Elmondani viszont, hogy mire is nyílik ezáltal lehetőség, már borzasztó nehéz. A program minden eleme a koherens egész részeként létezik, „tudja" tehát, mire való. Ezeket az összefüggéseket azonban emberi nyelven megfogalmazni egy kívülálló — felhasználó — számára, a legtöbb fejlesztő erejét meghaladja.

Még a számítástechnika hőskorában kialakult egyfajta szakmai tolvajnyelv, amely a „vájtfülűek" egymás közötti kommunikációjának céljára kiválóan megfelelelt (s amelyet jobb híján a laikus nagyközönség is nagyrészt átvett). Igen ám, de megjelent a színen az „elefánt a porcelánboltban", az imádott és átkozott felhasználó. Ha azt mondták neki, file, a file botját sem mozdította, ha bájtokról esett szó, rögtön egy kiló bájt óhajtott volna, a hard meg ugyanúgy verte, mint a szoft. Ezeknek? — kérdezte az egyszeri számítástechnikus. Igen, ezeknek — döntött a fejlődés, döntött a piac. És „ezeknek" illenék egyre inkább emberi nyelven közvetíteni mindazt az információt, amelyre szükségük van, s amelyek forrása többnyire külföldön, idegen nyelveken fakad.

Lehet, hogy eljön az idegen nyelvekről „szakmagyarra" történő fordítás után a szakmagyarról magyarra fordítás ideje is? Lehet, hogy mégiscsak a magyartanár lesz a jövő egyik kulcsfigurája? Meglátjuk. Amikor szerzőinknek mostani számunk vezértémáját vázoltuk, a fenti gondolatokat állítottuk a középpontba.

V. J.

Nyelvhelyesség-ellenőrzőre várva...

Az agglutináció visszavág

Nagy fába vágják a fejszéjüket, akik meg akarják valósítani a magyar nyelvhelyesség-ellenőrző programot. Pillanatnyilag három független társaság dolgozik rajta. Ha csak részeredményeket érnek el, de már kielégítően használható termékkel szolgálnak, munkájukat akkor is szép sikernek lehet elkönyvelni. Mindenesetre nagy szükség van titkárnőket, újságírókat és más tollforgatókat („billentyűpergetőket") segítő eszközre, amely elősegítené, hogy nyelvtani szempontból hibátlan írások kerüljenek az írások „címzettjeinek" szeme elé, s e cél (jó közelítéssel) megvalósíthatónak tűnik.

Az egyre könnyebben megszerethető szövegszerkesztőkkel együtt jár az igények fölbukkanása és fokozódása a nyelvhelyesség-ellenőrzők iránt. Természetessé vált, sőt a titkársági teendőkkel kapcsolatos szövegszerkesztési munkákhoz hozzá is tartozik, hogy a géppel készített üzleti levelek, értesítések nyelvtani szempontból helyesek legyenek. Sajnos a magyar nyelvhez — legalább az alapvető szabályokat érvényesítő, a legfontosabb elvárásokat kielégítő — „spelling checker" mind a mai napig nem létezik.

Sokféle nyelvhelyesség-ellenőrző program létezik a világon. A legegyszerűbbek csupán azt vizsgálják, hogy a szövegben előforduló szóalakok léteznek-e az adott nyelvben, annak rögzített szótárában. Ennél többet tudnak azok, melyek nemcsak jelzik a feltehetően hibás szóalakot, hanem felkínálják a kijavítás alternatíváit is. A még bonyolultabb programok megkísérlik a mondat nyelvtani szerkezetének vizsgálatát, értelmezését is.

Az ellenőrzők mindegyike szótárra épít. Régebben a szótárak kötöttek voltak, idővel azonban kiderült, hogy szükség van arra, hogy a szótárat a felhasználó maga is bővíthesse. Ennek oka a különböző témák differenciált szókészlete, s hogy minden embernek megvan a maga szókincse, amely ugyanúgy azonosítja a személyt, mint az ujjlenyomat.

Az egyes szavakat az emberek másmás gyakorisággal használják. A névelők, a kötőszavak, a személyes névmások mindenféle szövegben nagyon gyakran szerepelnek; de ha például egy újságíró egy évben 10 000 különböző szót használ, akkor ennek mintegy negyedét csak egyszer-egyszer írja le. Tapasztalatból tudjuk, hogy valamely diszciplína szakirodalmából összeállított 5000 szavas szótár az azonos szakterületen publikált újabb cikkek szavainak legfeljebb 10 százalékát nem tartalmazza. Másik tudományterülettel foglalkozó cikkekben viszont esetleg 50 százalékot is elér-

INFORMÁCIÓKÉRÉS: 43 ▼

het e szótár szerint ismeretlen szavak aránya.

Egy jó, a számítógépi szövegszerkesztéshez illeszkedő szótárnak tehát eleve elég nagynak kell lennie, és ugyanakkor bővíthetőnek is. Mi lehet az oka, hogy míg elsősorban az angol, de a német, francia stb. nyelvekre is már több mint tíz éve kereskedelmi termék a nyelvhelyességi ellenőrző, addig a magyar nyelvre csak mostanában jelennek meg hasonló szoftverek, s minőségük elmarad attól, ami az említett példák alapján elvárható lenne. Pedig a magyar programozók nem ügyetlenek; többször bizonyították, hogy képesek komoly problémákkal is megbirkózni.

Kemény fában jól fejlett göcsörtök

Közismert, hogy a magyar nyelv nem egyszerű ragozó nyelv, hanem úgynevezett agglutináló. Ez azt jelenti, hogy a szövegekben előforduló szóalakok szótőből és a rájuk több rétegben tapadó toldalékokból állnak. Ha a szótárban szerepel a kutya szó, akkor elvárható a programtól, hogy felismerje a kutyám, kutyául, kutyáinkat stb. szavakat. Hasonló az igény például az angol nyelvhelyességi ellenőrzőknél is. A foot mellett kell hogy legyen feet, a dog mellett dogs. A lényeges különbség, hogy míg egy átlagos angol szó legfeljebb 4-5 alakban szerepelhet a szövegben, addig egy átlagos magyar szó több ezer formája helyes. (Összehasonlításul: a latin nyelvekben 100 alatti, a szláv nyelvekben néhány száz az egy szóhoz tartozó különböző szóalakok száma.)

Ebből a szempontból az angol az egyik legszerencsésebb nyelv, hiszen a főneveknek lehet egyes és többes számú alakjuk, a mellékneveknek alap-, közép- és felsőfokuk, az igéknek pedig 5-6 különböző formájuk. A francia és a spanyol nyelv igeragozásánál ennél jóval gazdagabb a toldalékolás, de még a szláv nyelvek névszó- és igeragozása sem olyan gazdag, mint a magyar nyelv toldalékolása. Míg az angol nyelvben azt a módszert követik, hogy minden szónak összes ragozott alakját — akár kézi úton is — a szótárba tuszkolják, addig ez az út a ragozó nyelveknél nem járható. Az ilyen nyelveknél a szavak mellé ragozási osztályozást is kódolnak, és ez meghatározza, milyen toldalékok követhetik az adott szót.

Az ilyen osztályozási módszer kitűnően működik azoknál a nyelveknél, ahol a szó végére csak egyetlen esetragot vagy igeragot kell elhelyezni. Nem ez a helyzet a magyarban, ahol a végződések nagy része után újabb végződés jöhet, tehát azt is meg kell mondani, hogy a képzett szó ezután milyen ragozási osztályba esik.

Itt a piros, hol a piros?

Tehát a magyar nyelvhelyesség-ellenőrző program kidolgozását nyelvészeti munkának kell megelőznie, ami sajnos nem olyan egyszerű, mint más nyelveknél. Mivel a magyar nyelvben a ragozás és a szóképzés rendkívül gazdag, számos olyan problémával kerülünk szembe, melyek a legtöbb nyelvben fel sem merülnek. Ezeket a nehézségeket sajátos módszerekkel kell megoldani, s nem lehet más nyelveknél bevált kaptafára dolgozni. A továbbiak felsoroljuk azokat az alapkérdéseket, melyek megoldásra várnak a magyar „spelling-checker" kidolgozásánál, ezek részletes kifejtésére következő lapszámunkban térünk vissza. Beszélni fogunk arról, hogy milyen toldalékokat kell figyelembe venni, hol és hogyan kell alternatívákat keresni, s arról is szólunk majd, hogy mikor van és mikor nincs helye a kompromisszumnak nyelv- és számítástechnikus között.

Naszódi Mátyás

INFORMÁCIÓKÉRÉS: 30 ▲

Nyelvünk nyelvet nyújt

Helyzetjelentés a végekről

A saját anyanyelvéről mindenkinek van valamilyen elképzelése, de ezt általában nem alapozza meg tudományos érv, hanem elsősorban érzelmi színezetű. De még a hivatásos nyelvészek is csak feltételezésekből indulnak ki, amikor elkezdenek egy-egy részterülettel foglalkozni. Nekik, és a laikusoknak is van egy járható út feltételezéseik ellenőrzésére s arra, hogy anyanyelvünkről helyes képet nyerjenek: adatokat kell gyűjteni róla, és azok, illetve az elődök által összeállított gyűjtemények alapján vonni le következtetéseket. Az adatgyűjtés régen elkezdődött a nyelvészi körökben, de napjainkban a számítógépesítés elterjedésével újabb távlatok nyíltak meg a téma előtt.

A számítógép előtti időkben az adatgyűjtés keserves kézi munka volt. Egy-egy nyelvész, például a híres „Erdélyi szótörténeti tár" szerzője, a néhány éve elhunyt Szabó T. Attila évtizedeken át gyűjtögette céduláit, amelyekre adatait feljegyezte, utána kezdődött az adatok rendezése, csoportosítása. Nagy ugrás volt, amikor Papp Ferenc és társai a cédulák tartalmát már lyukkártyákra jegyezték és számológéppel dolgozták fel. Később mások magukat a szövegeket vitték számítógépbe, és ezekből származtattak információkat. Akárhogyan is volt, az adatok előkészítése idő- és pénzigényes munkát jelentett.

Napjainkban nagyszerű lehetőségek nyíltak meg a nyelvstatisztika előtt, hiszen az újságok, könyvek jó része eleve számítógépbe íródik (legalábbis a nyomtatás során), másrészt a Recognita és a hasonló programok néhány óra alatt egész regényeket képesek átalakítani nyomtatott eredetiről számítógépes, kódolt formára. Az, hogy a nyersanyag eleve számítógépben van, természetesen igen nagy segítség, de a munkának csak a fele. A nyelvész — miközben céduláit írogatta — kapásból megmondta, hogy a kérdéses szó milyen szófajú, milyen végződéseket kapott, milyen szerepet töltött be a mondatban, és így tovább. Hacsak nem valami egészen egyszerű betű- vagy szóalak-statisztikát csinálunk, az előkészítésnek ezt a fázisát manapság sem úszhatjuk meg. De már van választási lehetőségünk: vagy mégis az ember végzi az előkészítést, természetesen változatos variációjú számítógépes segítséggel, vagy valamilyen program. Most egy ilyen munkáról számolunk be, melynek eredményei sok szempontból érdekesek lehetnek szakmánk köreiben.

Mi természetesen a második úton indultunk el, nevezetesen egy programmal preparáltatjuk ki a szöveg szavait.

A módszerrel a legnagyobb probléma, hogy a preparáló programot nem lehet olyan tökéletesre csinálni, hogy utólagos ellenőrzésre ne legyen szükség. Például a program nem tud mit kezdeni a kétértelműségekkel (még akkor sem, ha az ember számára a szövegkörnyezet azonnal egyértelművé teszi a jelentést). Előnye viszont, hogy roppant elfogulatlan, és ott is észreveszi a határozatlanságokat, ahol nekünk az fel sem tűnik.

A magyar nyelv ragozó jellege

A magyar nyelv úgynevezett agglutinatív nyelv. Ez azt jelenti, hogy a szavak végére több rétegben végződéseket ragaszt. Az ilyenkor idézni szokott példa, hogy „elkelkáposztásítottalanítottátok" természetesen csak vicc, de a valóságban azért akadnak hasonló szavak, például: „meg-szent-ség-telen-ít-het-etlen-ség-é-t" vagy „meg-term-ékeny-ít-het-etlen-ség-é-nek". A statisztika igen érdekes képet mutat.

Próbaképpen kivettünk három cikket az Alaplap korábbi számaiból, és megnéztük, hogy melyik szó hány végződést kapott:

	1. szöveg		2. szöveg		3. szöveg	
0-végződés:	1668	59,94%	2813	62,00%	1384	62,01%
1-végződés:	839	30,15%	1282	28,26%	659	29,53%
2-végződés:	246	8,84%	397	875%	162	7,26%
3-végződés:	27	0,97%	43	0,95%	27	1,21%
4-végződés:	3	0,11%	2	0,04%	0	0,00%
5-végződés:	0	0,00%	0	0,00%	0	0,00%
Összesen:	2783	100,00%	4537	100,00%	2232	100,00%

Korábban egy 12 000 szavas mintában 4 olyan szót találtunk, amelyiknek 5 toldaléka volt. Persze a dologban rejtőzött egy adag csalás, mert mi csak ragokat, jeleket és néhány alapvető képzőt néztünk meg. Ha félretesszük a középiskolás tankönyvet és elővesszük valamilyen modernebb nyelvészeti munkát, ott ezekre a fogalmakra a következő meghatározásokat találjuk.

Rag = Olyan végződés, amely mindig a szó legvégén áll. Két fajtája van. Az igerag igékhez illeszthető. A névszórag (esetrag) névszókhoz. Az adott szófajra minden korlátozás nélkül alkalmazhatók.

Jel = A jelek meghatározott sorrendben a ragok előtt, de a szótő, illetve a képző után állnak. Jele van a birtokviszonynak, a többes számnak és a hiányzó birtoknak (például: Pistáé). A jelek korlátozás nélkül illeszthetők a helyükre, és a jel után is szabadon alkalmazhatók a ragok, de egy szóban egyfajta

jelből legfeljebb egy szerepelhet (kötött helye miatt).

Képző = A képző a képzetlen vagy már képzett szónak megváltoztatja a jelentését, átviszi egy másik szófajba, újabb vonzatokat rendel hozzá. Több képző is jöhet egymás után, de meg kell előzniük a ragot és a jeleket. A szótól függ, hogy alkalmazható-e rá a képző és hogyan változtatja meg az értelmét.

A képzők nagy része nem viselkedik szabályosan, hiszen másfajta kapcsolat van az asszony és az asszonyság, a katona és a katonaság, a király és a királyság között; nem is szólva arról, mennyire kétséges, hogy például a programozóság szó — nem mesterkélt gondolati környezetben — egyáltalán létezik-e.

Persze nem minden képző ilyen deviáns viselkedésű. Hiszen például az új felfogás szerint a melléknév fokozása is képzős szerkezet, mivel mindig a rag és a jelek előtt áll, tehető utána újabb képző (például: kisebbít), a középfokú melléknévnek keletkezik egy -nál/-nél ragos vonzata (minél kisebb); ugyanakkor lényegében minden melléknévből képezhető középfok. Hasonló a helyzet a melléknévi igenevek képzőivel, az ás/-és képzővel, az -ú/-ű képzővel stb.

Ezeket a jó képzőket mi reguláris képzőknek hívjuk. A program alapvetően csak a szabályos viselkedésű végződéseket tudja kezelni: azokat, amelyeknél meg tudjuk mondani, hogy milyen szavakra alkalmazható az illető képző, hogy miként származtathatjuk az új szó jelentését a régiből, továbbá, hogy milyen ragozási osztályba sorolható a keletkezett szó, és — amennyiben ismerjük a régi szó vonzatait — melyek lesznek az új szó vonzatai.

A fenti statisztika tehát azt mutatja, hogy a program hány reguláris viselkedésű végződést talált a szövegben. A rossz viselkedésű képzőkről semmit nem mond. Igen nehéz valamit is leszögezni bizonyos szavak esetén: vajon az „egyik", „egyszerű", „egyébként" szavakat képzett szavaknak tekintsük-e? Hasonlóan kérdéses a „rögzít" vagy „értékadás" szó is, hiszen rögz vagy rögöz szó, illetve értékad ige nincsen. Persze a két dolog nem egészen ugyanaz, hiszen olyan szabály mindenesetre van, hogy amikor egy igéből a cselekvés folyamatát leíró főnevet képezünk, akkor elékapcsolhatjuk a tárgyrag nélküli tárgyat (például: értékadás). Tehát ez egy teljesen szabályos képzés (szóösszetétel). De most akkor ez egy szó vagy két szó? A nyelvészek ilyenkor rendszerint megnézik, hogy mit mond az adott szóról az értelmező szótár, és aszerint járnak el.

Miután a fenti statisztika elég stabilnak látszik ennél a három szövegnél és a további megvizsgáltaknál (ezekben a kapott százalékok nem mutatnak nagy ingadozást), próbaképpen vegyünk ki mondjuk ezer szót valamelyik szövegből, majd nézzük meg, milyen további (nem reguláris) végződéseket találunk bennük; és fogadjuk el, hogy az arány a többi szövegnél is hasonló.

A 2. szöveg első ezer szavát vizsgálva a következő eredményeket kaptuk:

szóra alkalmazható bármely képző, esetleg az új szót bizonyos képzőkkel már nem módosíthatjuk. Ez azonban nincs így, hiszen vannak olyan szavak, amelyekben egy képző többször is előfordul (például barátságosság). Tehát elvileg a szóképzés ciklusba eshetne, mégsem teszi, mert határt von az emberi felfogóképesség. Mesterkéltnek érezzük azokat a szavakat, amelyekben ötnél több végződés van, és elkerüljük őket.

	Reguláris	Összes végződés
0-végződés:	586	533
1-végződés:	315	331
2-végződés:	90	114
3-végződés:	8	19
4-végződés:	1	3
Összesen:	1000	1000

A fentiekből az látszik, hogy a reguláris végződéseket sokkal gyakrabban használjuk a nem regulárisoknál, és csak kevésbé módosítják az arányokat. A szavak több mint fele nem kap végződést, több mint negyede egy végződést kap, kb. nyolcada kettőt, százból néhány kap hármat, ezerből néhány négyet, tízezerből néhány ötöt, ennél több csak elvileg létezik.

Hány szavunk van?

Az első kérdés: mi az, hogy szó? Legalább két dolgot kell megkülönböztetni: a szóalakot — azaz a szót, ahogy le van írva — és a lexémát, vagyis a szót, miután levágtuk róla a ragokat és jeleket, esetleg bizonyos képzőket is. Az algoritmus ilyenkor az szokott lenni, hogy addig vagdossuk a végződéseket, amíg a szó jelentése nem változik meg. Mindez nagyon szubjektív dolog, még ha az irányelveket ennél sokkal precízebben is írjuk le.

Egy lexémából persze rengeteg szóalak alkotható. Egy igének mintegy 260 ragozott alakja van. A legtöbb igére ezenkívül alkalmazható a -hat/-het, -tat/-tet, -gat/-get képző is. Továbbá minden igéből képezhető főnév és melléknévi igenév, ezekre azután alkalmazhatók a főnevek és melléknevek toldalékai. Egy névszóból, ha csak a ragokat és jeleket alkalmazzuk, mintegy 1200 alak állítható elő. De természetesen képzők is vannak a világon.

Mivel az igéből névszó és névszóból ige képezhető, elvileg egy szóból végtelen sok másik származhat. Ennek persze gátat szabhatna, hogy nem minden

Sok-e az említett öt vagy kevés? Egy angolnak vagy németnek biztos sok. Ezekben a nyelvekben a szó végén általában egyetlen végződés van, de sokszor még az sincs. Az orosz nyelvben az előtag és egy vagy két utótag a szokásos eset. A finn és török nyelvészektől viszont azt hallani, hogy náluk nyolc végződés sem ritka egy szón. Akárhogyan is van, az angolok megtehetik, hogy a szótárban nemcsak a szót tárolják, hanem annak toldalékolt alakjait is. A magyarban vagy a finnben ez elképzelhetetlen.

Elvileg tehát rengeteg szavunk van, de vajon mennyit használ egy ember? Meglepően keveset! Múltkoriban — feldolgozva egy 80 oldalas tanulmányt — mindössze 4345 szóalakot talált a program. Egy másik alkalommal 10 különböző szövegből (a kábítószercsempészetről szóló újsághírtől kezdve a számítógépes levelezőrendszer kézikönyvéig), melyek összesen mintegy 40 oldal terjedelműek voltak, 12 937 szóalakot kaptunk. Az 1989-ben megjelent Füredi-féle gyakorisági szótár, melyből később még bővebben merítünk példákat, 508 008 szónyi szöveget vizsgált kétszáz szerzőtől, és 91 471 szóalakot különböztetett meg, és ezek 33 169 lexémához tartoztak.

Egy ember egy témakörben viszonylag kevés szóval él. Juhász Gyula összes, azaz 1373 versében 11 606 lexéma szerepel. Arany János összes versében, ami 271 000 szót tesz ki, 23 500 lexéma van. A Petőfi Sándor összes műveiben (tehát a leveleket, cikkeket, műfordításokat is ideszámítva) található 448 309 szó 22 719 lexémát tartalmaz.

Persze a fenti adatokban jelentős bizonytalanság is maradt, hiszen kézi módszerekkel készült a statisztika, és a különböző szerzőknél eltér a metodológia is. De hogy mégis van benne valami, ez összevetésekkel valószínűsíthető: más kutatások eredményei szerint például Shakespeare 24 000, Victor Hugo pedig 28 000 különböző szót írt le összes művében. A fentiekből arra szoktak következtetni, hogy egy ember kb. 20 000 szót használ aktívan (és persze ennél sokkal többet ért meg passzívan, de hogy mennyit, nem tudhatjuk, mert erre nézve nincsen írásos bizonyíték).

Ezzel szemben Vargha Dénes, aki tudományos témabeszámolók szavait dolgozta fel számítógéppel, eddig mintegy 240 000 lexémát gyűjtött össze. Látható tehát, hogy a nyelv szókészlete azért hatalmas, mert mindenki a saját szókincsének birtokában beszél: más nyelvet alakítottak maguknak a méhészek, a nyelvészek, a gépészek, a kertészek stb. Ha van egy elemző programunk, amelynek 5000 szavas szótára van (elsősorban) a mesterséges intelligencia témaköréből, akkor ennek alapján érdekes dolgokra deríthetünk fényt. Tapasztalati ténnyé válik, hogy ha egy ilyen tárgyú, nagyobbacska cikket megvizsgálunk vele, akkor körülbelül a szavak 10%-át nem leli meg a szótárban, de ha egy orvosi cikket elemeztetnénk, akkor ez az arány valahová 30 és 50% közé esne.

Mi is úgy becsüljük, hogy kb. 20 000 lexéma lehet a különböző területek művelői által részlegesen elkülönített nyelvek közös magja. A többi olyan szó, amelyet csak az emberek egy csoportja ismer. Természetesen a tudományban és a technikában rengeteg idegen szó van, de vajon nem lehetne legalább olyan jól érthető igazi magyar szavakkal az, hogy „besztekkeljük az adatot" vagy éppen „kipoppoljuk"? Szebben is hangzik talán az adatokat „elvermelni" és „kipiszkálni" — hiszen az adott szövegkörnyezetbe ilyenkor sem az értelmező szótár meghatározásai szerint passzítjuk a szavakat, tehát a szakmán kívül álló ezt ugyanúgy nem érti, mint ha az előbbi, kétségtelenül nyelvrondító, „magyarosított" angol kifejezésekkel kommunikálunk.

Kedvenc szavaink

Az egyes szavak természetesen nem egyforma gyakorisággal fordulnak elő a szövegekben. És persze a szövegtől is függ, hogy melyik szó milyen sokszor szerepel benne.

Az imént említett forrásmű, a két éve kiadott munka (Füredi Mihály — Kelemen József: A mai magyar nyelv szépprózai gyakorisági szótára) 200 írónak 258 művéből indult ki. A szerzők véletlenszerűen kivett 508 008 szónyi szöveget vizsgáltak, és a leggyakoribb szavaknak a következőket találták:

a, az (névelők)	10,99%
és (kötőszó)	1,76%
nem (tagadószó)	1,70%
van (a létige és alakjai: lesz, vóna stb.)	1,61%
hogy (kötőszó)	1,42%
az (névmás)	1,08%
is (kötőszó)	1,02%
ez (névmás)	1,01%
de (kötőszó)	0,77%
egy (névelő)	0,70%

A mellé írt százalékok azt jelentik, hogy a szövegben előfordult összes szónak hány százalékában szerepelt az illető lexéma. A sorolásból látható, hogy az összes szó 11%-a névelő, a leggyakrabban használt tíz lexéma adja a szavak 22,07%-át; a könyv további adatai szerint az első ötven a szavak 34,50%-át, az első száz pedig a szavak 41,24%-át. Az első száz szó zöme kötőszó és névmás; csupán 14 főnév, 15 ige és 4 melléknév (jó, nagy, kicsi, szép) szerepel az élbolyban.

Mi is megnéztünk ilyen szempontból néhány tudományos-technikai szöveget. Azokban is ugyanezek a szavak és szófajok dominálnak, csak a főnevek mások. Az irodalmi szövegben az első 14 főnév nagy része ugyanis testrész és rokon megnevezés volt. A technikai szövegekben ezek szinte egyáltalán nem szerepelnek. Az igék között gyakoriak a „tud", „kell", „akar", „szeret" jellegű segédigék.

Érdekes továbbá megjegyezni, hogy a Petőfi műveiből való szavak alapján összeállított szótár anyagát is sorba rakták gyakoriságuk szerint, és ott is lényegében ezek a szavak jöttek ki, azonban a második helyen nem az „és", hanem az „s" szó áll, nyilván verstani okokból.

A szófajok megoszlása

A szófajokról igen érdekes statisztikákat lehet készíteni, mindössze az a kérdés, milyen szófajok vannak, továbbá, hogy az ezekhez való tartozást miként állapítsuk meg. Kérdés például, hogy a melléknévi igenevek a melléknevek közé kerülhetnek-e. A válasz lehet igen — hiszen a melléknévi igenév is lehet jelzője vagy állítmánya a mondatnak —, és lehet nem, mert van olyan mondat, amelyben az egyik megállhat, a másik nem. Például az „Alma eladó." helyes mondat, de az „Alma piros." nem. Hasonlóan: abban a mondatban, hogy „Ez a probléma már régen megoldott." az utolsó szó melléknévi igenévnek minősül, és a nyelvészek egy része szerint a mondat magyartalan; míg a következő mondatban: „Ez a probléma már régóta megoldatlan." az állítmányt általában melléknévnek tartják; bár a két képző viselkedése teljesen párhuzamos.

Mivel az általunk lefolytatott vizsgálatoknál sok szó szófaji besorolását az elemző algoritmus adja meg (kiszámolva a szótő és a képző kódjából), így a szófajokra kissé más adatok jönnek ki, mint a korábbi statisztikákból, ahol a szófaj meghatározására az volt az eljárás: „Ha kétséges, nézd meg az értelmező szótárban, ha ott nem szerepel, akkor a helyesírási tanácsadó szótárban, ..." — stb.

Bizonyos egyértelmű szófajoknál végül is nincs gond, a két eredmény eltérése bőven tolerálható:

	Nálunk	Gyakorisági szótár
Névelő	12,86%	11,76%
Igék	18,93%	17,57%
Főnevek+fvi névmások	29,86%	26,94%
Melléknevek	8,99%	6,96%

Az eltérés a határozószóknál és a módosítószók esetén feltűnő. A probléma

nyilván abból fakad, hogy másképp ítéljük meg a „mögé", „alul", „során" és hasonló szavakat. A különbség azonban nemcsak közöttünk van, hanem általában a magyar szótárak között, hogy mi mikor határozószó, mikor igekötő, mikor névutó, mikor ragos főnév stb. Mint például Komlósy András — 8 szótár összehasonlításával — megállapította: mindenütt igekötő a „fent", egyeseknél a „bent" is, de a „lent" sehol sem.

A kétértelműségekről

Mivel a magyar ragozó nyelv, három különböző oka lehet annak, ha egy szó kétértelmű: lehet a szótő kétértelmű (például „Hova nyúl a nyúl?"), lehet a végződés kétértelmű (például „ő törik" vagy „őt törik"), és lehet a szónak több felbontása tőre és végződésre (például „Érte érte el.") Néha egy szónak több felbontása is van. Nekünk a következő számok jöttek ki: 11 725 szóhoz 15 005 felbontást találtunk.

Egyértelmű	8827	(75,3%)
Kétértelmű	2549	(21,7%)
Háromértelmű	320	(2,7%)
Négyértelemű	24	(0,2%)
Ötértelmű	4	(0,0%)

A legtöbb többértelmű szó egytagú és ragtalan vagy egy toldalékja van. Minél több végződést kap egy szó, annál valószínűbb, hogy a szó egyértelművé válik, hiszen más a „nyúlok", „nyulak", „nyúlik". A többféle felbontás viszont a közepesen hosszú szavakra jellemző.

Másik oldalról viszont a szövegkörnyezet is egyértelműsítheti a szót. Ha a szó előtt egy „a" névelő áll, akkor a szó nem lehet ige (például „a nyúl"). Ezzel együtt a leírt szövegek egy része természetüknél fogva kétértelmű. A múlt idejű melléknévi igenév az esetek igen nagy részében azonos a múlt idejű igével, ezért nem lehet bizonyos mondatokról eldönteni, hogy cselekvő vagy szenvedő szerkezetűek-e. (Például „Az ügyes programozó Nyugaton is keresett." Ha ez a mondat cselekvő, a programozó tényleg ügyes.)

A toldalékok eloszlása

Egy szövegben (csak a reguláris végződéseket számolva) a toldalékok zöme rag. 34,07% esetrag, 16,68% igerag. A jelek száma kb. 20%. A maradék 30% képző. A képzőknek több mint a felét a melléknévi igenevek képzői és az „-ás/-és" teszi ki. A többi képző aránya egy, két, fél százalék.

Az esetragok megoszlása sem egyenletes. A leggyakoribb a tárgyrag a „-t" 31,70%. A következő csopotba tartozó ragok 8 — 9%-ban fordulnak elő, ezek a -nak/-nek, -val/-vel, -ban/-ben, -on/-en/-ön, -an/-en. A következő csoport ragjai 2%-nál többször fordulnak elő, ilyenek -ba/-be, -ból/-ből, -nál/-nél, -kor. A többiek csak tört százalékban fordulnak elő.

A második csoport ragjaira az jellemző, hogy rendkívül sok és különböző célra használjuk őket. A -nak/-nek raggal például a következő kapcsolatokat jelölhetjük:

Pista levelet küld Bélának.	Részeshatározó
Pista a barátja Bélának.	Birtokos
Bélának kell otthon maradnia.	!Alany
Béla nekiment az asztalnak.	???
Bélát kiképzik tűzoltónak.	???

Számos esetben nem tudjuk megokolni, hogy ott miért éppen azt a ragot kell választani. Ilyenkor valamiféle idiómaszerű jelenség a „bűnös". Régebben ennek egyes eseteiben szó volt képes helyhatározóról, de nemcsak a helyragok kapcsán van ilyen jelenség, hanem minden rag vonzatában.

Konklúziók

Cikkünkben a magyar szövegekben szereplő szavakról és végződésekről összegyűjtött adatainkat ismertettük. Meg kell mondanunk, hogy ebben a munkában maga az adatgyűjtés viszonylag kis energiával elintézhető volt. Az igazi fáradozást az jelentette, amíg rájöttünk, hogy a szükségesnek látszó adatokat hogyan lehet összeszedni, mit lehet a feladatokból géppel megcsinálni, mit pedig egyszerűbb kézzel, hol vannak elvi problémák és hol technikaiak. Közben nagyon sokat tanultunk a magyar nyelvről, részben a tudományos irodalomból, amelyet felhasználtunk, részben a kapott eredményekből; nem kis részt a kudarcokból, amikor kiderült, hogy egy-egy jól hangzó definíciót miért is nem lehetett alkalmazni a programban.

Tudjuk, hogy a magyar nyelv mondatai nem akárhogyan egymás mellé rakott szóalakokból épülnek fel. Azt szokták mondani, hogy a magyar nyelv szabad szórendű nyelv, amelyben a nyelvtani viszonyokat a toldalékokkal fejezzük ki. Ezt azzal illusztrálják, hogy a „Pista Marit szereti." mondat szavait bármilyen sorrendbe lehet permutálni. Hogy ez a dolog nem lehet egészen igaz, azt már a statisztika is mutatja, hiszen a szavak több mind felének nincs is toldaléka. Azonfelül, ha a „három fehér papírlap" kifejezést póbálnánk meg permutálgatni, kevés értelmes dolog sülne ki belőle. Egy következő cikkben azzal fogunk foglalkozni, hogy milyen szintaktikai kapcsolatok vannak a magyarban, és azok gépi felismerésével mennyiben érthetjük meg nyelvünket.

Farkas Ernő

Hol szűk a „díszmagyar”?

Tükörferdítés

Sok nemzetnek beletört már a bicskája abba, hogy saját nyelvére akarta átültetni a számítástechnika eredeti angol nyelvét. Érthető a törekvés, hogy mindenki anyanyelvén futtathassa programjait és alkalmazza a számítástechnikai fogalmakat, kifejezéseket. Ennek kivitelezése azonban a jelek szerint nem látszik reálisnak.

Vannak országok, ahol a számítástechnika „honosítását” elég diktatórikusan oldották meg, és megkövetelik a nemzeti nyelv kizárólagosságát. Aki franciából fordít, annak rémálmai közé tartozik egy-egy ottani számítástechnikai szakszöveg. A franciák ugyanis nyelvvédelmi megfontolások alapján minden (!) számítástechnikai kifejezésre azonnal francia szót alkotnak. Főhet a jámbor olvasó feje, de még a szakemberé is, vajh mit akar mondani a szerző. Sokszor maguk a franciák sem értik a saját anyanyelvükön publikált szakanyag egyes „újfrancia” kifejezéseit. Ez persze kiváló ösztönzés arra, hogy mindenki titokban elolvassa az adott témával foglalkozó szakirodalmat — angol eredetiben.

A németeknél a helyzet kissé más. Ők időnként szószörnyetegeket szülnek, de azt német precizitással, legtöbbször szolgaian lefordítva az eredeti angol kifejezéseket. Ha tehát a német szakember a tekervényes összetett szót nem érti meg, akkor viszonylag egyszerű a dolga: vissza kell „tükörferdíteni” angolra az inkriminált kifejezést, és utána minden megvilágosodik.

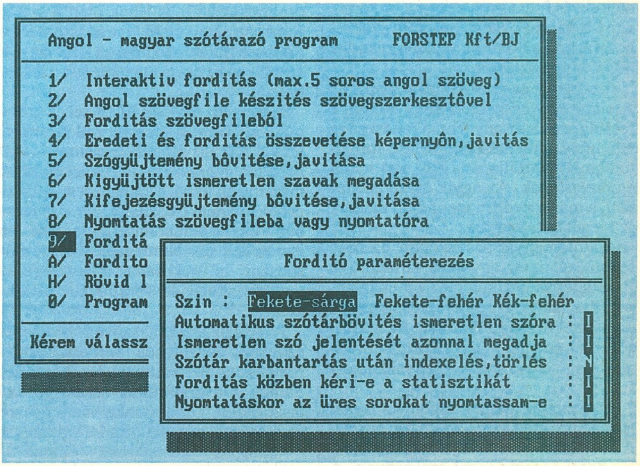
Kicsiny országunkban a két véglet — a minden áron való magyarítás és a kötőszavakat is angol eredetiben (legfeljebb mucsai kiejtéssel) megőrző szakzsargon — között húzódó széles skálán sokféle szaknyelvi törekvésnek lehetünk tanúi.

Ez a téma megérdemelne egy nyílt, részletes vitát, bevonva az egész számítástechnikai szakmát és a nyelvészeket is. A CWI Számítástechnika szerkesztőségének például több kezdeményezése is volt az angol szakkifejezések magyar változatainak megtalálására, sok jó és sok vitatható ötlettel. Ők vezették be például a port kifejezés honosítására a porta kifejezést. De a portot vezérlő integrált áramkört már nem merték portásnak nevezni. Ha a bus fogalmat adatsínre egyszerűsítjük (ami legalább egyértelmű), akkor az ezt irányító IC — a busvezérlő — talán váltóőr vagy masiniszta lehetne. A honosítások az újabb nyelvekre történő fordításkor számos leiterjakab forrásai. A németek például a csatlakozóra kitaláltak egy összetett szót: Schnittstelle. Ezt azután fordítóink mint vágólapot ültetik át magyarra. A grafikus programokban persze használjuk magyarul a vágólap kifejezést, de annak semmi köze a csatlakozóhoz.

A gyakorlat az értelmes magyarítási törekvéseket viszont igazolja. Jobb az egér, mint a mouse. Ugyanígy elterjedt az alaplap (naná – a szerk.) a mainboard helyett. Lassan a meghajtó is a drive fölé kerekedik. A magyarítást ki lehet találni, de nem lehet kierőszakolni a befogadását, használatának elterjedését. Ahogy kudarcot vallott a sci-fi helyett sokáig erőltetett fanti, nem mindig hullanak termékeny talajra a számítástechnikai kifejezések megmagyarított változatai sem. Ennek okát sokszor csak találgatni lehet, hogy talán a magyar szó jóval hosszabb az eredeti kifejezésnél, vagy a szakemberek nem tartják elég szabatosnak, egyértelműnek, pontosnak.

Nyelvérzékünk, a nyelvhasználat mindennapi gyakorlata dönt végül is a befogadásról, de úgy érezzük, hogy az Alaplapnak is feladata alternatívákat felkínálni egy-egy fogalom magyar kifejezésére, s ha azok egyikét sem képes integrálni nyelvünk, akkor az egyértelműség azt követeli, hogy vissza-visszakanyarodjunk a forráshoz, a rendszerint angol eredetihez. Ilyen megfontolásból népszerűsítjük például a közprogram és a szabadszoftver kifejezéseket a shareware, a freeware és a public domain helyett, tudva azt, hogy az eredeti kifejezések jelentésének minden árnyalata nem mindig menthető át csorbítatlanul a magyarba.

Kj.



Szótártesztek

Egy szó mint száz...

A számítástechnikai eszközök fejlődése felkínálja a lehetőséget: fordítás közben mentesítsen bennünket a számítógép minél több rutinmunkától, például a szavaknak a szótárból való kikeresésétől. Ezzel munkánk alaposan felgyorsulhatna. Ugyanakkor meg is nő az étvágyunk: mi lenne, ha a géppel legalább nyersfordítást végeztetnénk? Nagy volt a kísértés e kihívás elfogadására a fejlesztők részéről, és mutatóba több részleges eredmény is született, de ezek alkalmazhatósága elég kezdetleges.

Itthon is növekvő számban versengenek egymással ilyen-olyan szótárprogramok. Ezek közül választottunk ki három, Magyarországon már sok helyütt alkalmazott programot.

Nyelvtanból: elégséges

A szegedi Forstep Kft. programozója, Balogh János matematikus készítette el a cég által forgalmazott Info rendszert. Ez egy Foxbase+-alapú adatbázis, amely többek között a Clipperre és egyéb programokra vonatkozó információk mellett egy fordításra is képes angol–magyar szótárt tartalmaz, a Trans programrendszert.

Nagy várakozással tekintettem a teszt elé: számítástechnikai szakszövegből kiindulva mennyire lesz elfogadható a nyersfordítás, mely ennek a segítségével készül. A program másolásvédelem nélküli, és futtatásához vagy a Foxprun, vagy a teljes Foxbase program szükséges. (A kód a Foxbase saját tárgykódja, .FOX kiterjesztéssel.)

A program egyszerű ötleten alapszik. Amikor szövegszerkesztővel behívjuk feldolgozásra az anyagot, a rendszer az állomány elmentésekor dBase-formátumba konvertálja azt. Egy sort tekint egy rekordnak. Az egyik mező az eredeti, a másik pedig a megfelelő magyar szöveget tartalmazza.

Angol–magyar szakszótára hasonló felépítésű, igen egyszerű struktúrájú. Az ötletet valószínűleg az a munkamenet adta, ahogyan egy nyelvet nem tudó ember nekifog a fordításnak. Ez lehetőséget ad arra is, hogy a korrekciót viszonylag egyszerűen megvalósíthassuk.

A tesztre az általam különben is etalonnak tekintett vírus-adatbázis normál ASCII szöveges formáját választottam. Ez 8854 sornyi, angol szakszöveg. A program hozzám eljutott változatának egyik lemezén szerencsére egy olyan szövegszerkesztőt találtam — a QEditet —, mely a feladathoz illeszkedő installáció után az EMS-t és a merevlemezt felhasználva, képes jókora állományok kezelésére. Ezzel töltöttem be, majd mentettem el a szöveget. A program ekkor létrehozta a mintegy 3 megabájtnyi nyers adatállományt. Utána elindítható volt a fordítás.

A rendszer a kiakadásáig 6255 sort fordított le, és ebben mintegy ötszáz ismeretlen kifejezést — jobbára rövidítést vagy nevet — talált, amelyet részben az ismeretlen szavak adatbázisában, részben pedig a fordítás folyamán kézzel meg kellett adni. Úgy tűnik, a programnak nem kenyere a nagy szövegmennyiség, mert ismételt futtatás után is ugyanazon a helyen akadt ki... Az eredendő ok rendszerhiba, azaz programozási hiba lehet, mert a szótár felfrissítésére utaló parancssornál a Foxbase futtató modulja az állományok szabályos lezárása után kilépett a DOS-ba.

A teszthez 4 Mbájt EMS-sel és matematikai segédprocesszorral ellátott Exact AT állt rendelkezésemre — 40 Mbájtos gyors merevlemezzel és EGA monitorral. Tehát a kiakadást memóriaprobléma nem okozhatta. Az első esetben 4 óra 50 percet futott a program. Az ismeretlen szavak megadása után, másodszor is nekifutva a szövegnek, ugyanez a futásidő 2 óra 50 percre csökkent...

A rendszernek a szöveg javítására szolgáló, kétablakos szerkesztőprogramja megfelelő munkalehetőséget ad. Sajnos a nyersfordítás minősége katasztrofális. Több munka van a készre fésülésével, mint ha újra lefordítanánk — közepes angoltudással — az anyagot. Az angol nyelvtant csak annyira ismeri, hogy az egyes szóalakokat, ha eltérő nyelvi toldalékok vannak mellettük, ismeretlenekként újból megkérdezi.

A fordítás minősége csak részben javul a szótár feltöltésével. A program a magyar nyelv szabályaiban kiábrándítóan kevéssé igazodik el. Ennek ellenére mint az első és egyetlen PC-s kísérletet az automatizált fordítás megvalósítására, mindenképpen méltányolni kell. Szintúgy, hogy képes az angol és a magyar szöveg párhuzamos kinyomtatására is. Igaz, ennek a képe meglehetősen kaotikus, de a sorok számozása sokat segít a dolgon. Hosszasabb kutatással, a magyar és az angol nyelv szabályainak jobb algoritmizálásával ez a munka alapját képezheti későbbi, sikeresebb fejlesztéseknek.

Talán meg lehetne kísérelni azt az utat, amelyet az Európai Gazdasági Közösség fordítórendszerének kidolgozásakor követtek: nem közvetlenül fordítottak, hanem közvetítő nyelvként a rendhagyó ragozástól és kivételektől mentes műnyelvet, az eszperantót választották. Ilyenkor az algoritmizálás feladata jelentősen leegyszerűsödik: a kivételektől és az egyéb, a számítástechnikus elméjét zavaró tisztátalanságoktól mentes nyelv és az élő nyelv kapcsolatát kell csak algoritmizálni... Szurkolunk, hogy valaki megpróbálkozzék ezzel a módszerrel is.

Tanulni muszáj!

A nyelvet tanulókat célozta meg a Datamanager, amikor elkészítette Szótár nevű programját. A nagyon szép, kulturált kiállítású programcsomag 2.3 verziószámú változatát volt módunkban megnyúzni.

A program _D-H.DBF Dbase-formátumú alapadatbázison nyugszik. Hibájául róható fel, hogy más programokhoz képest nagyon kevés szót és kifejezést tartalmaz. A szótári modul alapvető hiányossága, hogy nem lehet betölteni tárban maradó módon más szövegszerkesztők alá. Így az a felhasználó, akitől magam is elkértem a teszthez a programot, fordításkor két gépen szimultán dolgozik, egyiken van ez a program, a másikba pedig szövegszerkesztővel csépeli bele az anyagokat.

A program kellemesen kezelhető, még olyanok számára is, akik csak alkalmazói szinten ismerik a számítástechnikát. Az alapadatbázis nem bővíthető, de a magunk kívánsága szerint akárhány saját adatbázist kialakíthatunk. Ha ezt a tulajdonságot nézzük, akkor a program kifejezetten jó a jelenlegi szótárprogramok között. Örömünket mégis lerontja néhány kezelési logikátlanság. A legbosszantóbb: ha keresés üzemmódban lapozni akarunk a szótárablakban, erre nem a megszokott iránybillentyűket, hanem — állandó tornagyakorlattal — az F5 és F7 funkcióbillentyűket kell használni. (Lehet, hogy tudatosan készült így?)

Ha egy adatállományt indexelni szeretnénk, akkor egy gyalogtetű sebességére számíthatunk: mintegy kétezer szóval 20 másodperc alatt végzett... s mert közben egy rövid életű feliraton kívül semmilyen visszajelzést nem ad, a felhasználónak az az érzése, hogy a gépe kiakadt.

Kiválóan beválik viszont tanulópárként: tudásunk, szókincsünk visszakérdezésére. Itt is találunk következetlenségeket. Például ha egy jel angol nevét kérdezi, akkor nekünk mégsem annak a magyar nevével, hanem magával a jellel kell válaszolnunk... Kezelése azonban elég logikus:

— A segédinformációkhoz (help) mindig az F1 billentyűvel juthatunk el.

— A különféle hangjelzések az F2 billentyűvel kapcsolhatók be vagy ki. Sajnos a bevezető nóta semmivel sem tiltható le.

— A Szótár program adatbázisa kisbetűs, ékezetes magánhangzókkal készült és csak á, é, ü, ó, ú, í van benne.

— Az ékezetes betűk technikai okokból (az ASCII sorrend mechanikus átvétele miatt) az ábécé végére kerültek, ezért rendezéskor a „z" után először az „ü", majd az „é", az „ö", az „á", az „í", az „ó", s végül az „ú" következik. Ez egy kicsit talán megzavarja a felhasználót.

A főmenüben a kép alsó sorában megjelenő utasítások szerint lehet eligazodni; vagy a menüpontok sorszámával, vagy a kurzorbillentyűkkel adhatjuk meg a szükséges funkciót. Mindkét esetben az Enter lenyomásával fogadtatjuk el a kijelölést. Ekkor a jobb alsó sarokban egy almenü jelenik meg (kivéve a tömörítés funkciót), amelyből az előzőhöz hasonlóan választhatunk.

E szótárprogram az általunk összeállított szószedetet is kinyomtatja.

Összességében úgy értékelhető, miszerint fordítás közbeni munkára alig alkalmas, mindazonáltal professzionális szótárral állunk szemben. Helye ott lenne az oktatási intézmények nyelvi kabinetjeiben, természetesen rendesen feltöltve adatokkal. A szótár ugyanis magában rejti a lehetőséget, hogy nemcsak angol, hanem bármilyen, a hagyományos ASCII jelekkel leírható ábécéjű nyelv szótárát létrehozhassuk vele. A szótár adatszerkezete dBase-alapú, de a kezelő szoftverét más, magas szintű programnyelven írták. A szép kivitelű, másolásvédelem nélküli program cégreklámnak sem utolsó!

A közel ideális szótár: RDIC

Akinek az utóbbi években szüksége volt fordítási feladatai elvégzésekor valamilyen kétnyelvű szótárra, alighanem erősen megkötöttnek érezhette a kezét. Tárban maradó szótárként még a közelmúltban is mindössze egyféle volt kapható, s az is másolásvédett...

Piackutatás közben jutottam el Székesfehérvárra, s ott kaptam meg Kelemen Lajos programozó RDIC nevű, rezidens szótárának 1.9 verzióját. A program néhány backuplemezen került el hozzám, és gondot okozott az ezen lévő utasítások alapján az installálása. Később találkoztam intelligens, könnyen üzembe helyezhető változatával, mely kettő darab, 1,2 Mbájtos lemezen három, tömörített állományt tartalmazott.

A programot kötelezően a C: meghajtó DIC nevű alkönyvtárába kell bevinni, mert különben nem találja meg a help állományait. A szótármodulok az útvonal megadásával szabadon helyezhetők el, de csak az első, a főkönyvtár alatti szintig terjedő mélységben.

A szótárat egy sajátos és megdöbbentően rövid (mindössze 27 388 bájt hosszú!) tárban maradó adatbázis-kezelő program, az RDIC.EXE kommandírozza. A betöltéskor az útvonallal együtt meg kell adni a betöltendő szótár nevét. Jelenleg két állomány forog közkézen, mindegyik igencsak méretes: a Gerhun német–magyar és magyar–német szótár, míg az Angol a magyar–angol, angol–magyar szótárat tartalmazza. A fordítás iránya mindegyiknél menet közben is megcserélhető.

Ha a programot betöltöttük a megfelelő szótármodullal, az ALT és a bal SHIFT billentyűk együttes lenyomásával indítható. Ezt más programok futása közben is megtehetjük. A program használatához feltétel, hogy a monitor 80 x 25-ös alfanumerikus módban legyen. Ekkor a legtöbb szövegszerkesztő kifogástalanul működik vele, kivétel a Personal Editor néhány változata és a Norton Editor, amelyek a képernyőt direkt írással, assembler programból kezelik, és nem engedik a program aktivizálódását.

Az első aktivizálódáskor a program a menüt kínálja fel. Ez különben a program aktív állapotában az „m" betűvel is előcsalogatható. A program magyar interfésze a CWI kódkiosztást alkalmazza. Az F1 fukcióban a kurzort rá kell vinni a kiválasztott szóra a szövegszerkesztőben, majd az Enter gomb megnyomására a kurzorral ellentétes képernyőoldalon megjelenik annak és a hasonló alakú szavaknak a lehetséges jelentése.

Hibája ennek a szótárnak, hogy az így kapott és jóváhagyható jelentés nem vihető át közvetlenül a szövegszerkesztőbe. Ugyancsak problémás, hogy a szókészlet sem bővíthető. A nyelvek fordítási helyzete azonban pillanatok alatt megváltoztatható, csak az új adatbázis igényel egy kis odafigyelést.

Az F2 gombbal kereségélhetünk — akár szótöredékek megadásával is! — a szótárban. A szövegszerkesztőhöz az ESC gombbal térhetünk vissza. Amíg meg nem szokjuk, kényelmetlen, hogy a szótár nem ismeri a nyelvtant. Amit más hasonszőrű program tud, azt itt nekünk kell megtennünk, például visszamenve a szövegszerkesztőbe, egy szóköz beszúrásával csonkolni kell a szót úgy, hogy a program végül is megtalálja.

A program az eddig vizsgált ilyen jellegű rendszerek közül a leggyorsabb, és a tárban a legkevesebb komplikációt okozza. Ezt annak köszönheti, hogy a szerzője egy, az OS/2 adatbázis-kezelőjéről mintázott saját adatbázis-eljárást dolgozott ki. A szótárállományok önindexeltek; az egyes részek előtt az állományok tartalmazzák az indexeket, míg a fő indexcsoportok egy önálló, .JMP kiterjesztésű állomány ugrótáblájában vannak.

Összevetve: az RDIC a Magyarországon fellelhető szótárprogramok legjobbika. Valószínűleg a szerző akarata ellenére szabadszoftverként terjed. Nem ártana, ha az alkotó ebbe belenyugodva forgalomba bocsátaná a nyelvi csonkolást, a szótár és a szövegszerkesztő közötti kétirányú, interaktív kapcsolatot és a bővíthetőséget megteremtő, regisztrált változatot — elérhető áron, másolásvédelem nélkül. Ezzel valószínűleg szép karriert futhat be az RDIC „olcsó szoftverként". (Várjuk a szerző jelentkezését, akiről mindössze annyit tudunk, hogy valahol Szeged környékén él.)

Kis János

Szintaktika és szemantika

A gép (nyersen) már fordít

A természetes nyelvek számítógépes fordítása mindig is a mesterséges intelligencia kutatóinak egyik fő célkitűzése volt. A próbálkozások már sokszor kudarcba fulladtak, de újra és újra feléledtek, mert olyannyira fontos és izgalmas a téma, hogy mindig akadnak megszállott művelői. Jelenleg éppen sokadik reneszánszát éli, mutatva a lényeget: ami kell, az kell! A világ számos helyén foglalkoznak vele, és néhány éve megjelentek a piacon kapható, illetve bérbe vehető fordítóprogramok is.

A COLING'90 konferencia mellett rendezett kiállítás egyik látványossága a Fujitsu (japán) és az Aris (német) cég közötti kooperáció eredményeként megszületett Atlas II fordító volt. A japán szöveget latin betűkkel írták be, ez a monitoron már szótagírásban jelent meg, majd a mondat befejezésekor a program a szótagjeleket szójelekké vonta össze. Ahol ez utóbbi nem volt egyértelmű, a visszajelzés megállt, és a rendszer egy menüben az adott szó helyett másokat is felajánlott. A mondat begépelése után pár pillanattal megjelent a német szöveg, amelyet — most már a speciális igények szerint — tovább lehetett szerkeszteni.

A fordítás minőségét a következő számok jellemzik: a mondatok 20%-a teljesen hibátlan, 40%-ában, ha nem is tökéletesen, de megmaradt a mondat értelme, 25%-ában komoly hiba került a fordításba, 15%-ában a fordítás egyáltalán nem sikerült. A fordítóprogram teljesítménye kb. 1000 szó óránként (SUN-4 típusú munkaállomáson). Nagyjából ez jellemző a mai piaci fordítókra.

Átalakítás SYSTematikusan

A másik professzionális fordítóprogram, amelyet közelről kipróbáltunk, a Systran. Ez a program a francia Systran cég nagyszámítógépén fut, és telefonvonalon érhető el PC-kből (más országokból is). 10 nyelvpár között nyújt lehetőséget fordításra. A lefordítandó szöveg mellé három szakterületet kell megjelölni, és a rendszer ezen szakterületek szótárai alapján végzi a fordítást. A kisebb nyelvekhez egyelőre nincs még meg minden szakszótár.

A profi fordítóprogramnak PC-n futó része, a Convert program különböző szövegszerkesztő programokat ismer, így biztosítja az adatátvitelt oda és vissza — a szöveg formátumának megmaradása mellett. A fordítás adagonként halad, de rendkívül gyors: egyoldalnyi szövegnél az egésznek az adminisztrálása és az oda-vissza út egy-két percet, míg maga a fordítás mindössze másodperceket vesz igénybe. A szolgáltatás természetesen éjjel-nappal működik.

A fordítás itt is nyersfordítás minőségű, tehát gondos ellenőrzést és utánszerkesztést igényel. Elsősorban hivatásos fordítóknak, illetve lektoroknak jelent könnyebbséget. Nem mindegy viszont, hogy 24 órán keresztül rendelkezésünkre áll, és egy ismeretlen szöveg hozzávetőleges tartalmáról bármikor tájékozódhatunk. További előnye, hogy a gépi fordítás nem ismer igazi mennyiségi korlátokat, rövid idő alatt igen hosszú szöveget lehet áttenni egyik nyelvről a másikra, és vannak olyan helyzetek, amikor az ennyire gyorsan megkapható nyersfordítás komoly segítséget jelent.

Az akadémikus kutatások sem állnak sokkal jobban. Ezek általában egy-egy részterületen valamivel többet tudnak, de globális teljesítményük nem jobb, és kezelhetőségükben elmaradnak a közönséges, de professzionális fordítóktól.

„Tapodtat is alig"

Mi az oka annak, hogy a gépi fordítás 30 év után is csak itt tart? A válasz elég egyszerű: egyrészt az emberi nyelv sokkal komplikáltabb, mint amilyennek látszik, másrészt az egyes nyelvek jobban különböznek egymástól, mint ahogy azt a rutinos fordítók tudásuk birtokában érzik.

Néhány éve még komoly nyelvtanárok és több nyelvet beszélő emberek is valahogy úgy képzelték a fordítást, hogy a szavakat behelyettesítjük a másik nyelv szavaival, az eseteket a másik nyelv megfelelő eseteivel, majd esetleg néhány szórendi változtatást eszközlünk, és a fordítás máris kész.

Ez a szemlélet olyan közelálló nyelvek esetében is képtelenségnek bizonyult, mint a cseh és az orosz. Gondot okoz, ha olyan szó fordul elő, amelynek két különböző jelentése is van, tehát ha azt a szót a másik nyelvben két különbözőre is le lehetne fordítani. Bizonyos szavakat a egyik nyelv egyes számúnak, a másik nyelv többes számúnak tekint, az egyik nyelvben hímnemű, a másikban nőnemű — s mindez a hozzájuk kapcsolódó igékre és jelzőkre is hatással van. Gondoljunk bele: még egy közönséges nőnemű névmást sem lehet egyértelműen a másik nyelv nőnemű névmásába áttenni, hiszen lehet, hogy az ott nem nőnemű.

Még nehezebb a helyzet az egymástól távolabb álló nyelvek esetén, amikor bizonyos szavak a másikban egyszerűen hiányoznak, s ezeket valamilyen körülírással kell kifejezni. S éppen az egészen elemi és sarkalatos szavaknál vannak leggyakrabban ilyen különbségek. Például a magyarban nincs olyan ige, mely az angol „have"-nek vagy a német „haben"-nek felelne meg. Néhány nyelvből (például a latinból vagy a kínaiból) hiányzik az „igen" szó. Az oroszban ismeretlen a „van" szó, és vagy tízféle körmönfont szerkezettel helyettesítik ezt a ténymegállapítást, jobbára pedig egyszerűen kihagyják. S miközben viszonylag könnyen megoldható egy nemlétező szó körülírása, annál nehezebben, hogy egy körülírást visszaváltoztassunk egyetlen tömör szóvá.

Ahhoz, hogy egy szöveget egy másik nyelvre viszonylag jól lefordíthassunk, igen nagy mélységben fel kell derítenünk a benne lévő nyelvtani és jelentésbeli kapcsolatokat. A jelenlegi fordítóprogramok a szintaktikai kapcsolatokat elég alaposan felfejtik, de ezáltal még a szemantikus kapcsolatoknak

csak kis hányadára derül fény. Ezenfölül: nem elég egyetlen mondattal foglalkozni egyszerre, hiszen a minőségi fordításhoz a szöveg összes kapcsolatát fel kellene ismerni, méghozzá valamilyen előzetes gyakorlatot, tudásmodellt feltételezve.

Miért nehéz?

Az általunk ismert fordítóprogramok működése általában egy függőségi gráf mentén követhető. A függőségi gráf azt írja le, hogy a mondat szavai milyen relációkban vannak egymással.

A viszonylatok egy részét maguk a szavak írják elő, ezeket vonzatoknak nevezzük. Például az igéhez kapcsolódhat alany, tárgy és határozó, s az adott ige megszabja, hogy ezek közül melyiket muszáj és melyiket lehet valóban hozzákapcsolni. Pontosabban az igének csak a megfelelő vonzatokkal együtt van értelme, hiszen mást jelent az, hogy „jelöl valakit valamire", mint „jelöl valamit valamivel". Ezek a törvényszerűségek annyira a szavakhoz kötődnek, hogy a szótárban tároljuk őket.

Vannak azután olyan kötöttségek is, amelyeknek már sokkal nagyobb a szabadságfokuk. Egy főnév mellé gyakorlatilag minden melléknevet odatehetünk jelzőnek, attól nyelvtanilag nem lesz rossz a szerkezet, legfeljebb értelmileg válik abszurddá. Hasonlóan: minden mondatba, amelyben van ige, betehetjük, hogy az ige által kifejezett cselekmény hányszor valósult meg.

A mai fordítóprogramok első fázisa, az analízis éppen ezeket a kapcsolatokat deríti fel. Ezután a program nem az egyes szavakat fordítja át a másik nyelvre, hanem a kapcsolatokat alakítja át a másik nyelv megfelelő összefüggései szerint. A fenti példában a „jelöl" szót és argumentumait az első esetben valószínűleg egy „nominate for..." kapcsolattá alakítja, a másodikban pedig egy „mark with..." formára hozza (az angol nyelv esetén). Ehhez lefordítja a jelzős szerkezeteket is. Azonban ez sem olyan egyszerű, mint az első pillantásra látszik. A névszói kifejezés alkotórészeinek minden nyelvben megvan a maguk sorrendje, például „Bush elnök" angolul „President Bush", egyes nyelvekben tehát a név a foglalkozás jelzője, más nyelvekben éppen fordítva.

Mindazonáltal jobb-rosszabb szabályrendszerek állíthatók fel, hogy milyen összefüggéseket kell felderítenünk és azokat milyen célnyelvi összefüggésekké kell átalakítanunk. Nagy különbségek vannak a fordítóprogramok között abban a tekintetben, hogy a közbülső alakban látszik-e még a mondat szórendje, megmarad-e, ha a mondat szenvedő volt, nem vész-e el, hogy az igéhez kapcsolódó mondatrészek milyen ragokkal kapcsolódtak oda, avagy csak a szerepük van megjelölve. Biztos-e egyáltalán, hogy mi volt a tényleges ige, vagy csak a jelentése van valahogy megadva.

A gordiuszi csomó

A gépi fordítás megfelelő szintjének elérésében érdekeltek egyik kézenfekvőnek látszó ötlete, hogy fordítsuk le a szöveget, illetve annak mondatait valamilyen szemantikus reprezentrációba, majd ebből a reprezentációból alakítsuk a közlést a másik nyelv szerint helyesen és egyértelműen. Ez az eljárás abból a szempontól is előnyösnek látszik, hogy ha több nyelvről gyakorta kell ugyancsak több nyelvre fordítani, akkor nem muszáj bármely két nyelv között megcsinálni az átvitelt, hanem elegendő minden nyelvből a közös közbülsőre konvertálni, és onnan kell a visszafordítást megcsinálni. Ezért ezt a módszert a többnyelvű fordítók jószerével mind alkalmazzák.

Az is igaz viszont, hogy ha a mondatot egy nyelvtől független reprezentációba alakítjuk, ez valamilyen mértékű információvesztéssel jár. Különösen az egymáshoz közeli nyelvek esetén jelent ez problémát, ahol egyes nyelvi konstrukciókat szinte változatlanul lehetne átvinni. Ezért életképes az olyan iskola is, mely arra esküszik, hogy nem szabad a szöveget nyelvfüggetlen formára alakítani, hanem csak a benne szereplő összefüggéseket kell kideríteni, majd ezeket a kapcsolatokat kell transzformálni a másik nyelv megfelelő konstrukcióivá.

A hagyományos fordítás legnagyobb előnye, hogy az ember — a fordító — nagyon rugalmasan alkalmazkodik, egyes helyeken csaknem szó szerint adja vissza a szöveget, más helyeken teljesen elrugaszkodik a leírtaktól, és csak a szöveg értelmét adja vissza. A számítógép jelenleg ilyesmire képtelen. A tapasztalat megerősíti, hogy ez nemcsak stilisztikai kérdés, hanem az értelemzavaró hibák keletkezésének magyarázata is.

Nem könnyű „visszaütni"

Szomorú, hogy míg szerte a világban mindenütt végeznek ilyen irányú kutatásokat — például a szomszéd országokban is —, addig nálunk e téren semmiféle érdemi munka nem folyik.

Szerencse, hogy a természetes nyelvű interfésszel kapcsolatos kutatásaink során valamennyire feltérképeztük a magyar nyelvű mondatok szerkezetét. Ez elősegíthetné, hogy a magyar nyelvről való fordítást viszonylag hamar megcsináljuk. Ha olyan rendszerbe tudnánk beilleszkedni, ahol a fordítást úgy értelmezik, hogy a mondatot valamilyen standard szemantikai reprezentációvá alakítják át, majd ebből generálják a célnyelvi mondatot, nem lenne rossz esélyünk. (Ez a generátor jellegű program már többé-kevésbé készen is van.)

A magyar nyelvű mondatok generálása már jóval hosszadalmasabb munkának látszik. Tudjuk ugyan, hogy mit kell beletenni a mondatba, de nem rögzíthetjük, hogy milyen sorrendben álljanak a szavak. A magyar nyelvről ugyanis köztudott, hogy szabad szórendű, de ez nem azt jelenti, hogy a szavak bármilyen sorrendben állhatnak, sokkal inkább azt, hogy nem algoritmizálhatjuk, mitől jó az egyik mondat, és mitől rossz a másik. Például:

Árpád a klaviatúrához ült.
Árpád ült a klaviatúrához.
* Ült Árpád a klaviatúrához.

? Árpád a terembe belépett.
Árpád belépett a terembe.
Belépett Árpád a terembe.

Keze ökölbe szorult.
* Keze szorult ökölbe.
* Szorult keze ökölbe.

*(A *-gal megjelölt sorrendek hibásak, a ?-lel megjelölt sorrend helyessége kétséges.)*

Az már világos, hogy a szavak jó sorrendje nem azon múlik, hogy mi az alany, mi az állítmány, hanem a nyelv egy másik rétegén. Ennek a rétegnek a szabályai valószínűleg olyan dolgokat írnak le, hogy melyik az új vagy a hangsúlyozott információ, hogy mi a visszahivatkozás, hogy a mondat kijelentés, kérdés, kívánság-e stb. Ezekből a szabályokból jelenleg elég keveset ismerünk, pedig nélkülük lehetetlen helyes mondatokat előállítani. Úgy látszik, hogy az angol nyelvben, ahol a szavak sorrendje kötött, ezek a szabályok kevésbé fontosak, de ott sem elhanyagolhatók, és a gépi fordítás számára nem megoldottak, amit az is mutat, hogy a számítógép egyelőre náluk is csak nyersfordítás szintű végeredményt tud produkálni.

Farkas Ernő

Mindent és azonnal!

Szinkrontolmácsolás — Japánban

Ma már a magyar számítástechnikusok számára is elérhető valóság az elektronikus levelezés a világban szerte dolgozó kutatók között. A távolsági korlátokat lényegében áthágta vagy legalábbis döntögeti a számítástechnika. De hiába hozná helybe például egy csodálatos számítógépes hálózat egy távol-keleti adatbázis adatait, aligha értenénk meg belőle egyebet, mint a statisztikai táblázatok adatait, esetleg a matematika, fizika, kémia ott is érvényes nemzetközi jelöléseit.

Ma még álomnak is túl szép volna számunkra egy olyan közvetlen kapcsolat — mondjuk Tokióval —, hogy japán barátunk anyanyelvén gépelje be üzeneteit, és azok számunkra érthető módon, angolul vagy ne adj' isten, magyarul érkezzenek el hozzánk, és megfordítva, a mi mondandónk úgy jusson el Tokióba, hogy közben átalakul japán nyelvű közléssé.

Mindnyájan érezzük, hogy magának az adatátvitelnek az időigénye eltörpül akár az üzenet keletkezésének időtartamához képest, akár a várakozásra elfecsérelt időhöz viszonyítva is. Amíg a számítógépes rendszer arra vár, hogy nyelvi formát öltsön az újabb kérdés vagy válasz bármilyen gyors ész- és kézjárású ember ujjai alatt, azalatt el lehetne végezni sokmillió számítógépes műveletet. Ebbe az időbe sok minden beleférne. Folyamatosan, szavanként lehetne követni már a szöveg elkészültét is, és menet közben kellene keresni a szavak megfelelőjét a másik nyelvben. Sok jelentés esetén válogathatnánk közülük az addig leírt szavak lehetséges jelentésének vagy használati körének figyelembevételével. Fel lehetne deríteni a szavak nyelvtani és jelentéstani kapcsolatait...

Egyszóval hihetőnek tűnik, hogy talán kompletten, végleges formára is le lehetne fordítani az üzenetet. Problémás esetekben (mondjuk elírások vagy szókihagyások esetén) rögtön vissza is kérdezhetne a számítógép, mielőtt esetleg eltorzulva és tovább torzítva továbbítaná az üzenetet. (Erre gondoltál? Nincs itt valami tévedés?) Vagy olyankor, amikor a megfogalmazás nem egyértelmű, menüszerűen fölkínálhatná a szóba jöhető értelmezéseket. (Segíts választani, melyik a helyes ezek közül.)

Akiknek megadatott...

Három és fél évvel ezelőtt a genfi V. Telekommunikációs Világkiállítás résztvevői már belekóstolhattak abba, milyen érzés lenne, ha a számítógép szerepe a kommunikációban nem állna meg a fizikai távolság leküzdésénél, hanem a különböző nyelvi kultúrák távolságát is csökkenteni tudná.

A jegyzőkönyv tanúsága szerint a következőképpen zajlott le egy műholdon át továbbított beszélgetés a kiállítás egyik amerikai résztvevője és egy tokiói fiú között. A könnyebb érthetőség kedvéért zárójelben a magyar fordítást is közöljük — ettől eltekintve betűhíven követjük az eredetit. (Mint látható, a jegyzőkönyv hűen megőrizte az elütések nyomait is. Ez azért tanulságos, mert egyúttal igazolja a rendszer „masszív" jellegét is: apróbb hibák nem zavarják meg a helyes értelmezést.)

— My name is Takeda. Please tell me your name.
(Takeda vagyok. Önt hogy hívják?)
= Hello, Takeda. My name is suzanne.
(Üdvözlöm, Takeda. Én Suzanne vagyok.)
— Where do you live?
(Hol lakik?)
= I live in geneva, but I come from California.
(Genfben lakom, de Kaliforniából jöttem.)
— I see. Have you visited Japan?
(Értem. Volt már Japánban?)
= Yes, but when I was 12 years old.
(Igen, de akkor még csak 12 éves voltam.)
— Please tell me the impression of this machine.
(Mi a véleménye erről a gépről?)
= Very interesting, quick and useful !
(Nagyon érdekes, gyors és hasznos.)
— Thank you.
(Köszönöm.)
= How many languages do you speak, Takeda?
(Hány nyelven beszél, Takeda?)
— I can speak only Japanese.
(Csak japánul beszélek.)
= That is ok.
(Nagyszerű.)

A fenti beszélgetés egy kétirányú (angol—japán, japán—angol) online gépi fordítórendszer imponáló nemzetközi bemutatkozása volt. A Toshiba cég kutatólaboratóriumában kifejlesztett rendszer nagy sikert aratott, és érthető módon mély benyomást gyakorolt a konferencia résztvevőire. A képernyőn természetesen a műhelytitkokból is sokat meg lehetett jeleníteni: követhetők voltak a fordítás egyes fázisai, és a kívánt részletességgel ellenőrizni lehetett az egyes lépések megalapozottságát a működés közben. (Érdemes belegondolnunk, hogy a beszélgetés „egynyelvűsítése" valójában kettős feladatot jelent: mást kell megjeleníteni a genfi beszélgetőtárs képernyőjén, mást a tokióién!)

Ne tanulj nyelveket!

Egy még hihetetlenebb teljesítmény elérése izgat mostanában sok japán kutatót és számos kutatóintézetet: nemcsak írásban, hanem valóságos, élő beszédben is megoldani a kölcsönös érthetőséget a normál párbeszéd megszokott válaszváltásával. A páratlanul ambiciózus terv megalapozásával kapcsolatban széles körű munka folyik a a „Basic Research of Automatic Translation Telephone" elnevezésű alapkutatási program keretében. A program gazdája és a kivitelezés fő letéteményese egy japán óriáscég, a Nippon Telegraph and Telephone Corporation (NTT), amely — érthető módon — a leginkább érdekelt egy ilyen teljesen újszerű, „nyelvfüggetlen" telefonkapcsolat felkínálásában. (Az NTT-ről érdemes annyit megjegyezni, hogy óriási tőkék bevonásával alig néhány éve privatizálták a céget. Méreteiről pedig fogalmat alkothatunk abból az egyetlen adatból, hogy részvényeinek összértéke meghaladja Amszterdam és Szöul tőzsdéjének teljes forgalmát.)

Az adott esetben azonban többről van szó, mint egy akármilyen hatalmas cég egyéni vállalkozásáról. Japán egyébként is híres arról, hogy erőforrásait rendkívül céltudatosan használja fel, és még a konkurens vállalatok között is az ésszerű munkamegosztás és a számunkra meglepően harmonikus összhang dominál bizonyos kulcsfontosságú kérdésekben. Mindez természetesen nem a véletlen műve: hozzáértő állami intézmények munkálkodnak azon, hogy „össznemzeti célkitűzések" elérésére (természetesen komoly állami fej-

lesztési alapok bevonásával) kialakítsák a vállalatok közös érdekeltségét. Ilyen módon már a nyolcvanas évek elején sikerült Japánban nemzeti üggyé tenni az 5. generációs számítógépek kifejlesztését is, elérve, hogy a japán cégek színe-java bekapcsolódjon a fejlesztésekbe.

Nos, japán megítélés szerint a „nemzeti ügy" kategóriába sorolható a fenti, ambiciózus célkitűzés is, hiszen megvalósítása az egész japán tudományos és üzleti élet számára páratlan lehetőségeket takar: szerves bekapcsolódást a világ kommunikációs hálózatába, és nem pusztán technikai értelemben, hanem a szó szoros értelmében — a nyelvi korlátoktól is megszabadulva.

Állami részről a Csúcstechnológiai Központ (Japan Key Technology Center, JKTC) volt a téma kezdeményezője, amely koncepcionális szakmai kérdésekben közös tudományos-technikai irányító szerve a Nemzetközi Kereskedelmi és Ipari Minisztériumnak és a Kommunikációs Minisztériumnak. A JKTC gondoskodik arról is, hogy ne legyen kétes a program anyagi fedezete: az adott témában — 70%-os állami részvétel mellett — 30%-ban vesz részt a magántőke a kutatások költségeinek finanszírozásában.

A téma alapkutatás mélységű megvalósítását hétéves időtartamra tervezték. Előzetes becslések szerint mintegy 17 milliárd yen (kb. 110 millió dollár) költség árán a 90-es évek közepére elérhető, hogy már iparilag is kialakítható legyen egy teljesen automatizált, világot átfogó szinkrontolmácsoló rendszer, amelyen keresztül telefonon szabadon társaloghatnak egymás nyelvét nem értő személyek is.

Vargha Dénes

A szótár

Nagy fába vágjuk a fejszénket! Alaplap voltunkhoz és lapkoncepciónkhoz minden szempontból méltó feladat, hogy lehetőség szerint még ebben az évben megjelenjék az Alaplap Könyvek sorozatban egy újabb Alap-mű: egy naprakész számítástechnikai értelmező szótár. Tudjuk, hogy ilyen típusú szótárból, szótárkísérletből több is van a piacon. De olyan egy sincs, amelyikre azt mondhatnánk: na itt van *a* szótár, ha ezt felütöm, *minden* kérdésemre — legyek laikus vagy a szakma veteránja — megbízható, kielégítő választ kapok.

Ilyen szótár a magyar piacon ma még nincs, lehet, hogy a teljes anyag nálunk is csak öt év múlva áll össze. Éppen ezért választottuk — Sebestyén Béla irányításával — azt a szerkesztési elvet, hogy először tematikus modulokként építjük fel és adjuk ki a szótárt. Megjelenik tehát az első kötet — mondjuk — az adatfeldolgozási terminológiát felölelve, a második a kommunikációra koncentrálva, a harmadik speciális szoftverterületek szókincsének értelmezését adva és így tovább. S egyszer, évek múltán — lefedvén az egész szakterületet — még egy nagy feladat marad: „összefésülni" az egyes szakterületeket *a számítástechnika* komplett szakszótárává.

S ha e sorok olvasói között akadnak olyanok, akik szívesen részt vennének egy „össznépi" szakszótár megalkotásában, kedvet érezve egy-egy szócikk megírására vagy más részfeladat megoldására, kérjük, jelentkezzenek a kiadó címén (Cédrus Kiadó, 1251 Budapest, Postafiók 71). Ígérjük, munkával — és reméljük, dicsőséggel is — el tudjuk látni őket!

NETREND

NETREND Általános Kereskedelmi és Szolgáltató Rt.

1089 Budapest VIII., Elnök u. 1. Telefon: 113-8217, 133-4760 Telefax: 113-9537

XT-10 számítógép	
– 640 kilobájt RAM	
– Multi I/O kártya	
– 360 kbájtos FDD	
– 101 gombos billentyűzet	29 800
– 83 gombos billentyűzet	3 700
– 101 gombos billentyűzet	3 800
AT 286-12/16 számítógép	
– 1 megabájt RAM	
– FDD/HDD vezérlő + S/P kimenet	
– 1,2 Mbájtos FDD + vezérlő	
– 101 gombos billentyűzet	38 950
AT 286-16/21 számítógép	
1 megabájt RAM	42 250
NEAT 286-16/21 számítógép	
1 megabájt RAM	46 250
NEAT 286-20/26 számítógép	
1 megabájt RAM	48 850
NEAT 286-24/32 számítógép	
1 megabájt RAM	57 950
NOVELL AT terminál:	
AT-286-12/16 MHz, 1 megabájt	
101 gombos billentyűzet, 1,2 megabájt	
floppy	
ARCNET K., 14" MONITOR	59 500
AT 386-20/25 számítógép	
– 2 megabájt RAM	
– FDD/HDD vezérlő + S/P kimenet	
– 1,2 megabájtos FDD	
– 101 gombos billentyűzet	85 100
AT 386-25/33 számítógép	
2 megabájt RAM	92 000
AT 386-25/43 (64 kilobájt cache)	
2 megabájt RAM	102 700
AT 386-33/58 (64 kilobájt cache)	
2 megabájt RAM	109 300
AT 486-25/117 (128 kilobájt cache)	
4 megabájt RAM	254 500
AT 486-33/147 (128 kilobájt cache)	
– 4 megabájt RAM	296 500
AT 486-33 EISA (128 kilobájt cache)	
– 4 megabájt RAM	414 500
Monitor csatolókártyák:	
Monochrom	1 700
Color	2 400
EGA 800x600	6 500
VGA 800x600	9 600
VGA 1024x768	11 800
VGA 1024x768	15 600
VGA 1024x768	16 500
Monitorok (14 inches):	
Egyszínű (borostyánsárga)	11 900
Egyszínű (papírfehér)	12 500
EGA	34 500
VGA (1024x768)	38 500
VGA MultiScan	48 500
Floppymeghajtók:	
360 kilobájtos	5 750
1,2 megabájtos	5 950

1,44 megabájtos	6 600
Winchesterek:	
21 megabájtos(MFM)	19 000
42 megabájtos(MFM)	24 800
82 megabájtos(MFM)	56 500
43 megabájtos(IDE)	22 400
88 megabájtos(IDE)	38 500
124 megabájtos(IDE)	46 500
210 megabájtos(IDE)	75 900
182 megabájtos	89 900
384 megabájtos	189 000
766 megabájtos	219 900
Szünetmentes áramforrások:	
UPS 400 VA NOV	39 900
UPS 550 VA	32 000
UPS 600 VA NOV	45 800
UPS 1 kVA	54 600
UPS 1,2 kVA NOV	98 500
UPS mon. kártya	7 500
Nyomtatók:	
FX-850	49 500
FX-1000	46 500
FX-1050	48 750
LQ-850	76 400
LQ-2500+	165 000
DFX-5000	183 500
DL 5600 (színes)	195 000
HP LASERJET III.	199 900
Memóriabővítő kártyák:	
286-2 megabájt	9 900
286-3,5 megabájt	10 800
386-2/8 megabájt	15 000
RAM-ok:	
4164-10	140
41464-08	290
41256-08	160
41256-06	280
44256-08	780
511000-10	750
511000-08	760
CSIMM/SIP modulok:	
256K *9-08 modul	1 790
1024K *9-08 modul	5 690
Co-processzorok:	
80287-10	14 000
80387-20	44 000
80387-25	52 800
80387-33	69 900
CHAYES COMP. modemek:	
2400 baud belső	10 900
2400 baud külső	14 500
2400 baud MNP-5	18 500
Telefax kártya adó-vevő	
9600/2400 baud	32 500
Egerek, scannerek:	
GM-6000 egér	4 950
Microsoft egér	16 000
GS-4500 handy s.	32 500
HP SCANJET PLUS	259 000

Plotterek:	
SEKONIC 450	115 600
HP 7475A A3	215 000
HP 7550A A3	450 000
HP 7570A A1	448 000
HP 7575A A1	569 000
HP 7576A A0	788 000
HP 7595B A0E	999 000
HP 7596A A0E	1 290 000
MUTOH 910E	1 380 000
ARCnet kártyák:	
8 bit LIN DATA	4 800
8 bit ZOT	5 900
8 bit SMC	9 800
16 bit LIN DATA	9 800
16 bit ZOT	12 500
ETHERNET kártyák:	
8 bit NE-1000	11 500
8 bit DE-100	17 900
8 bit DE-150	19 600
16 bit NE-2000	16 500
16 bit DE-200	22 500
HUB-ok:	
Passzív HUB E4	2 000
Aktív HUB (int 4)	7 800
Aktív HUB (ext 8)	15 900
AHUB+ARCnet k.	12 000
Aktív HUB E8	14 900
Kábel 5 m 2xBNC	1 500
BOOT-EPROM	2 000
Hálózati kiegészítők:	
ETH. Transceiver	35 000
Transceiver BNC	25 500
Transceiver UTP	32 000
Repeater (2 port)	92 500
Repeater (4 port)	148 000
BOOT-EPROM	2 500
Jogtiszta hálózati szoftverek:	
Novell NetWare	
ELS Level I	54 000
ELS Level II	139 000
Advanced	240 000
SFT II. V.2.15	344 000
386 V.3.0	499 000
386 V.3.1	539 000
Network Asyncron Communication	
Server Program	110 000
Asyncron Remote	
Bridge Program	29 600
DLink Lansmart op. rendszer	31 800
DLink Bridge Pr.	29 600
DLink Acs. Pr.	28 000
Remote Access	19 600
Screen monitor	15 000
LAPTOP-ok:	
LT-3400 (NEAT)	199 000
CP-8100 (386)	299 000

A Netrend Rt. a Novell Inc. hivatalos dealere!
Komplex hálózati rendszerfelügyelet, tanácsadás, hardver- és szoftverkarbantartás.
Komplett CAD, DTP, Arcnet-Ethernet hálózatok szállítása igény szerint.
Kérje részletes tájékoztatónkat!
Készpénzfizetés esetén 5% kedvezmény!
Önkormányzatoknak, tanintézeteknek, egészségügyi intézményeknek, tömegszervezeteknek 5% kedvezmény!
Áraink áfa nélkül értendők és hathónapos cseregaranciát tartalmaznak!

INFORMÁCIÓKÉRÉS: 51 ◄

KOREOGRÁFIA 20 TÉTELBEN

Az egér interfész

Bizonyára sok programozó szeretné, ha programjait egérrel is lehetne vezérelni. Ennek legegyszerűbb módja a Genius Menu Makerrel vagy hasonló programmal az egérműveleteket billentyűkódokká alakítani és a hagyományos módon feldolgozni. Hatékonyabb, de bonyolultabb a Microsoft Mouse Driver használata. Ennél alcsonyabb szintű egérkezelést már nem javasolunk, mivel a Mouse Driver legfőbb jelentősége, hogy elrejti az egér típusától függő eltéréseket. A Mouse Driver leírása megtalálható a Microsoft Mouse User's Guide for IBM Personal Computers vagy a Genius Mouse User's Manual kiadványokban részletesen, angol nyelven. (Tudomásom szerint magyarul még sehol sem.) Ennek funkcióit fogjuk most itt röviden ismertetni. A leírás az 5.01 verziótól fölfelé helytálló.

A mouse.com TSR program lefuttatásával installáljuk a drivert, amely az int 33H (int 51) interrupthívással érhető el. A hívás előtt a driver jelenlétét illik ellenőrizni. Ha int 33H 0-ra vagy egy IRET-re mutat, akkor nincs betöltve a driver. A kívánt alfunkciót az AX regiszterben kell megadni. Az egyes alfunkciók más-más regiszterekben megadott paramétereket igényelnek. A visszaadott értékek is alfunkciótól függő regiszterekben vannak. A funkciók listája és rövid leírása a következő:

0.: Reset Mouse Driver
Ez a hívás alaphelyzetbe állítja a drivert. A default értékeket az egyes funkcióknál soroljuk. A visszaadott értékek: AX=0, ha van egér, BX a gombok száma az egéren.
1.: Enable Cursor Display
Ennek hatására a kurzor megjelenik a képernyőn.
2.: Disable Cursor Display
Ilyenkor a kurzor nem lesz látható a képernyőn. Ez a default.
3.: Read Cursor Location
A kurzor helyzetének és a gombok állapotának lekérdezése. Visszaadott értékek: BX=gombok állapota, (0. bit: bal oldali gomb, 1. bit: jobb oldali gomb, 2. bit középső gomb; =1 lenyomva, =0 nincs lenyomva), CX=a kurzor vízszintes koordinátája, DX=a kurzor függőleges koordinátája.
4.: Set Cursor Location
A kurzor helyzetének beállítása. CX-ben kell megadni a vízszintes, DX-ben a függőleges koordinátát. A virtuális képméret általában 640 x 200, Hercules grafikus módban 720 x 348, nagyfelbontású EGA módban 640 x 350.
5.: Read Button Press State
Ez a hívás az egér helyzetét adja meg egy gomb legutolsó lenyomásakor. BX-ben kell megadni a lekérdezendő gomb számát. 0=bal oldali, 1=jobb oldali, 2=középső. A rutin AX-ben a 3. funkcióhoz hasonlóan visszaadja a gombok állapotát. BX megadja, hogy a legutóbbi hívás óta hányszor nyomták le a gombot, CX és DX pedig a kurzor vízszintes ill. függőleges koordinátáját adja a gomb legutolsó lenyomásakor.
6.: Read Button Release State
Ez a hívás az egér helyzetét reprodukálja egy gomb legutolsó felengedésekor. BX-ben kell közölni a lekérdezendő gomb számát. 0=bal oldali, 1=jobb oldali, 2=középső. A rutin AX-ben a 3. funkcióhoz hasonlóan ad információt a gombok állapotáról. BX által megtudhatjuk, hogy a legutóbbi hívás óta hányszor engedték fel a gombot, CX és DX pedig a kurzor vízszintes, ill. függőleges koordinátáját mutatja a gomb legutolsó felengedéskor.
7.: Define Horizontal Range
A kurzor vízszintes mozgását a kép egy részére korlátozza. Ha a híváskor a kurzor a tartományon kívül volt, akkor a tartomány szélére ugrik. A tartomány bal szélét CX-ben, jobb szélét DX-ben kell megadni.
8.: Define Vertical Range
A kurzor függőleges mozgását a kép egy részére korlátozza. Ha a híváskor a kurzor a tartományon kívül volt, akkor a tartomány szélére ugrik. A tartomány tetejét CX-ben, alját DX-ben kell megadni.
9.: Define Graphics Cursor
A grafikus kurzort két 16 x 16-os bitminta definiálja. A képernyőtartalomhoz a kurzor helyén az elsőt AND-elve, a másodikat XOR-olva kapjuk a kurzort. Az eredő hatás a grafikus módtól függ. A két 16 szavas bitmintát egymást követően kell elhelyezni a memóriában. Híváskor ES:DX a bitmintákra mutat. BX és CX a kurzor középpontját definiálja. Ez -16 és +16 közti érték lehet.
10.: Define Text Cursor
Szöveges módban kétféle kurzor használható: szoftver- vagy hardverkurzor. Ha BX=1, akkor CX-ben kell megadni a kurzor első, DX-ben pedig az utolsó sorának számát. BX=0 esetén CX és DX két maszkot tartalmaz. A képernyőn a kurzor alatti karakter CX-szel AND-elve majd DX-szel XOR-olva jelenik meg. CX és DX alsó bájtja a karaktert, míg felső bájtja az attribútumot módosítja.
11.: Read Motion Counters
Az egér mozgásának legegyszerűbb követésére szolgál. Két előjeles egész értéket ad vissza: CX-ben az egér vízszintes irányú elmozdulása a legutóbbi hívás óta; DX-ben az egér függőleges irányú elmozdulása a legutóbbi hívás óta. A pozitív előjel jobbra, ill. fölfelé irányuló mozgást jelent.
12.: Define Event Handler
Lehetőség van az egér mozgásának interruptos kezelésére is. A driver a CX-ben meghatározott esetekben hívja az ES:DX-ben definiált rutint. Az egyes bitek 1-es állapota engedélyezi a hívást a feltétel teljesülése esetén:

CX bit	Feltétel
0	Egér elmozdulása
1	Bal oldali gomb lenyomása
2	Bal oldali gomb felengedése
3	Jobb oldali gomb lenyomása
4	Jobb oldali gomb felengedése
5	Középső gomb lenyomása
6	Középső gomb felengedése

A default CX érték 0.

13. és 14.: Enable/Disable Light Pen Emulation
A 13. funkcióval bekapcsolható, a 14.-kel kikapcsolható a fényceruza-emuláció.
15.: Define Sensitivity
Ezzel a funkcióval lehet beállítani, hogy mekkora egérelmozdulás feleljen meg a képen 8 pont elmozdulásnak. Az egér elmozdulását 1/200 inchben kell érteni. CX a vízszintes, DX a függőleges elmozduláshoz tartozó értéket határozza meg. Alaphelyzetben vízszintesen 8, függőlegesen 16 lépés tartozik a képen 8 pontnyi elmozduláshoz.
16.: Disable Cursor in Special Range
A kurzor kijelzése egy ES:DX által mutatott négyelemű egész tömb által meghatározott tartományban kikapcsolható. A tömb elemei sorrendben: a tartomány, bal, felső, jobb és alsó széle.
19.: Define Double-Speed Threshold
Az egér gyors mozgatása a képernyőn megduplázhatja a kurzor mozgási sebességét. A küszöbsebesség DX-ben lépés/másodperc egységben adható meg. A default 64. A dupla sebességű mozgatás kikapcsolása DX=32767 megadásával lehetséges.

Pintér Gábor

Újabb erősítés az adatbázis-kezelők piacán

Magyarul beszél a Progress

A bostoni központú amerikai Progress cég európai vezetője februárban tartott sajtótájékoztatóján jelentette be, hogy a Rolitron cég lett a Progress magyarországi disztribútora. Közel 20 mérnök foglalkozik a Rolitronnál a Progresszel, és már dolgoznak a magyar nyelvű változaton. A fejlesztőrendszer természetesen angol marad, és csak a futtatható rendszer lesz magyar nyelvű, de ez a példa is mutatja, hogy a magyar piacon előbb-utóbb fölénybe kerülnek a magyarul beszélő adatbázis-kezelők.

A Rolitron a tervek szerint átveszi magyarországi alkalmazásra az Európában már jól bevált (banki, kereskedelmi, szállodai stb.) rendszereket. A komplett rendszereket úgy kívánják értékesíteni, hogy az általuk forgalmazott szoftver mellett más cégekkel együttműködnek a hardver beszerzésében is. A legolcsóbb fejlesztői rendszer már 150 000 forint körüli áron beszerezhető, egyszerűbb hardverkonfiguráció esetén a Run-Time rendszer körülbelül 30 000 forintos áron várható. Természetesen nem tekintik a Progress-t az egyedül üdvözítő megoldásnak, használatát elsősorban a nagyobb rendszerekhez ajánlják, hiszen egy gép esetében a Clipper vagy dBase használata olcsóbb.

Generációváltás

A Progresst fejlesztőrendszerként üzemeltető felhasználók az adatbázis-kezelőn túl egy sor más érdekességre is számíthatnak. Így bizonyára sokak kíváncsiságát kelti fel a Progress saját programnyelve, amelyben megtalálható a harmadik generációs nyelveknél megismert ciklusszervező utasítások többsége, kiegészítve adatbázis-kezelő utasításokkal, ezek illeszkednek az ANSI által bevezetett SQL szabványhoz. A programnyelv fájlszervezése hangsúlyozottan támogatja formátumozott képernyők, lekérdezési listák, menük gyors kiépítésének lehetőségét. Abban az esetben, ha olyan problémával szembesül a programozó, amit nem lehet vagy nem célszerű a Progress 4GL programnyelvében megoldani (például bizonyos hardver-hozzáférések), akkor külső eljárások hívásával operálhat. Jól illusztrálja ezt, hogy egy pénztárgép vagy vonalkódolvasó vezérlését végző C rutin CALL utasítás segítségével beágyazható egy Progress 4GL programba.

A negyedik generációs nyelvek megjelenése az adatbázis-alkalmazások növekvő mennyiségi és minőségi igényeit tükrözi. A harmadik generációs nyelvekhez (C, COBOL, Pascal) képest jelentősen kevesebb kód írására van szükség, ami növeli az applikációs programok hatékonyságát, és csökkenti a fejlesztéshez szükséges időt. A 4GL rendszerre költött pénz bizonyosan megtérül a programírás, módosítás, karbantartás során jelentkező költségmegtakarítás révén.

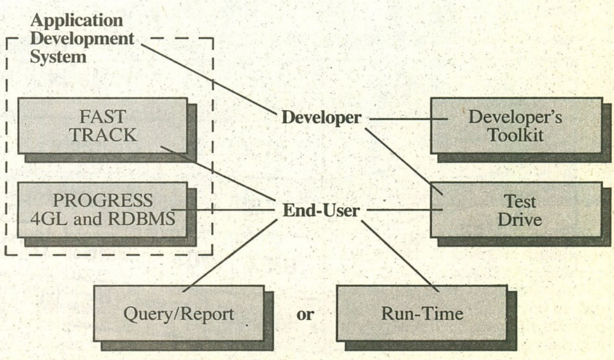
A Progress fejlesztőrendszer a Progress cég elsődleges terméke, mely tartalmazza a Progress 4GL-t és az RDBMS-t, a Fast Track-et és egy menüvezérlő applikációkészítőt.

A Progress 4GL és az RDBMS használata során tapasztaltabb feljesztők könnyen írnak egy- és többfelhasználós applikációkat a Progress 4GL angol szavakhoz hasonlatos utasításaival és az automatizált RDBMS-lehetőségekkel. Progress Fastereknél a fejlesztők felhasználhatják a képernyőfestő és menüvezérlő lehetőségeket a fejlesztés gyorsítására, a felhasználók viszont a listagenerátor és a formátumos lekérdező szolgáltatásait használhatják ki jó hatásfokkal. A Progress Run-Time alkalmas egy Progress-applikáció futtatásához ott, ahol nincs fejlesztőrendszer kulcsrakész alkalmazásként.

Angolhoz hasonló szavakat használva a 4. generáció automatizál sok olyan funkciót, amelyet csak lépésről lépésre lehet leírni részletesen egy 3GL nyelven, mint például C-ben vagy COBOL-ban.

Biztonságban az adatok

A modern adatbázis-kezelő rendszereknél a hardverfüggetlenségen túl a legfontosabb követelmény az adatbiztonság megvalósítása. Hálózatkimaradás, a hardver meghibásodása vagy akár programozói hiba sem eredményezheti a nagy munkával létrehozott adatbázisok megsemmisítését. A Prog-



ress által garantált adatbiztonság két fő pillére a következő:

— Automatikus adathelyreállítás a rendszer esetleges összeomlása után. Például hálózatkimaradás esetén a rendszer gondoskodik róla, hogy a még be nem befejezett tranzakciók ne kerüljenek végrehajtásra. Ilyenkor újraindítás után a Progress „törli" a félbemaradt műveleteket (pl. új rekord feltörése), így biztosítva az adatbázis integritását. Természetesen a legutolsó műveletek eredménye megsemmisül, de az adatbázis szerkezete és tartalma sértetlen marad.

— A második módszer a végrehajtott műveletek naplózásán alapszik: ez annyit jelent, hogy a Progress az adatbázist hordozó adattárolótól független lemezeken jelzi a legutolsó mentés óta végrehajtott műveleteket. Ezek után, ha az adatbázist tartalmazó winchester tönkremegy, nemcsak az utolsó mentés áll rendelkezésre, hanem a Progress a napló alapján automatikusan végrehajtja a lejegyzett műveleteket, helyreállítva a hiba bekövetkezése előtti állapotot. Nagy adatbázisok esetében ez többhetes munkától kíméli meg a felhasználót.

Az adatbiztonság mellett ma már szintén alapkövetelmény az adatvédelem, a hozzáférési rendszer korrekt megszervezésének lehetősége. A Progress rendszerében az adatbázis egészére és elemeire (táblák, mezők) is definiálhatók a hozzáférési jogok. Megadható, hogy mely felhasználók végezhetnek bizonyos műveleteket az adatbázison, és mely felhasználók használhatják az alkalmazói programot vagy annak egyes részeit.

Az integrált adatszótár használatával könnyű karbantartani az adatstruktúrákat. Egy editor érvényesíti az inputot a szótárbeli definíciókkal szemben, és ellenőrzi az utasítások szintaktikáját, rámutatva bármilyen hibára, és automatikus formázó generálja a képernyőket és a listákat úgy, hogy az eredmények közvetlenül láthatók legyenek. A Progress lehetővé teszi, hogy a fejlesztők C-ben írt rutinokat is hívhassanak, valamint azt, hogy használhatóvá váljék a bennfoglalt SQL. E rugalmas végfelhasználói interfész mögött egy relációs adatbázis-kezelő rendszer biztosít nagy tranzakciós áteresztőképességet, referenciális integritást, automatikus rekord-lokkolást és -helyreállítást a szerver-kliens architektúrát kihasználva.

A Progress programcsomagjába integrált relációs adatbázis-kezelő rendszer leveszi a fejlesztő válláról az adatbázis definiálásával, a memória- és lemezkezeléssel kapcsolatos problémák megoldásának terheit, így lehetővé teszi a feladat lényegi elemeire történő összpontosítást. Szintén nem igényel külön programozói munkát adatbázisok hálózati elérésének biztosítása, a 6.0-ás verzió lehetőséget nyújt osztott adatbázisok kezelésére is. A rendszer automatikusan kezeli több felhasználó egyidejű hozzáférését több — akár különböző típusú — gépeken elhelyezett adatbázishoz.

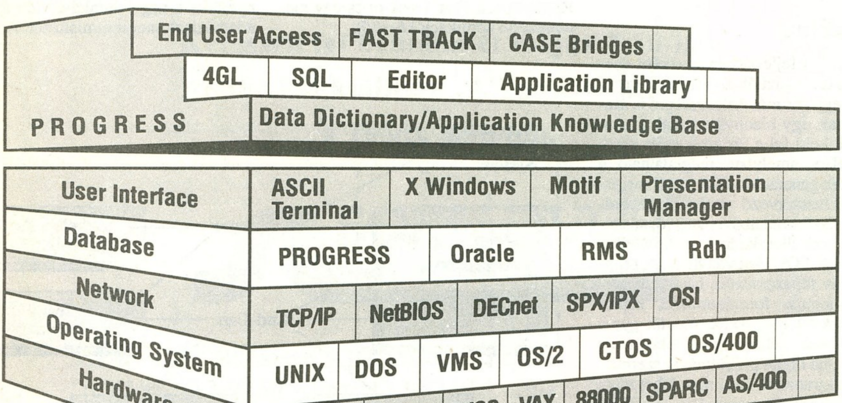
Progress-applikációk

A Progress-applikációk mini- és mikroszámítógépek százain futnak a következő operációs rendszerekkel: UNIX, ULTRIX, AIX, MS-DOS, OS/2, VAX/VMS stb.

Egy applikáció, amely az egyik környezetben lett kifejlesztve, változtatás nélkül átvihető egy másik alá. Például egy többfelhasználós rendszert ki lehet fejleszteni egy XENIX gépen, és utána átvihető egy VAX/VMS-re anélkül, hogy akárcsak egy forrássorhoz hozzá kellene nyúlni. Progress-applikációk szintén számos hálózaton képesek működni MS-DOS környezetben, de NetBIOS vagy a NOVELL IPX protokollját használva, UNIX-alapú LAN-ok általában a TCP/IP protokollt használják. VMS környezetben VAX gépek DECnet protokollal tudnak információt cserélni.

Gyakorlott Progress-fejlesztők számára a Fast Track az applikáció készítésének jelentős részeit meggyorsítja, így a képernyő, a lista tervezését és menük létrehozását. A Progress Run-Time egyetlen olyan részt sem tartalmaz, amely applikációíráshoz szükséges, így a Run-Time-felhasználók futtatni tudnak egy kulcsrakész alkalmazást, de nem tudnak egyetlen saját eljárást sem írni.

Sziebig Andrea



Kisokos v.1.1

Úton a pontos címhez

Ahol számítógép van, ott általában címanyagot is nyilvántartanak vele. Az számít kivételnek, ha valahol nincs ilyen igény. Viszont meglepetéssel tapasztaltuk, hogy a széleskörű alkalmazási lehetőség ellenére mennyire nincsenek a magyar sajátosságokhoz alkalmazkodó, igazán jó címtárkezelők, telefonregiszterek. Arra biztattunk tehát néhány fiatal programozót, hogy vágjanak neki a feladatnak. Így készült el a Kisokos program, amelynek demó változatát mostani számunk mágneslemez mellékletén is közreadjuk.

Tapasztalatainkat nem illik általános érvényű megállapítások formájában közölni, mert egyrészt nem biztos, hogy „mintasokaságunk" jól reprezentálja a teljes kört, másrészt a túlnyomó többségre érvényes elmarasztaló általánosítással is megsérthetünk olyanokat, akik a kivételhez tartoznak. Mindezek tudatában a címtárkezelés magyar sajátosságainak felsorolását azzal a minden pontra érvényes feloldással kezdem, hogy „Magyarországon sok helyen...":

1. Annyira nem törődünk a címlisták frissen tartásával, hogy az elhunytaknak is hosszú évekig küldözgetjük a tájékoztató anyagokat, árlistákat, tiszteletpéldányokat stb.
2. Annyira hiányzik belőlünk a pontos címzés megalkotásának belső igénye, hogy alig van olyan cím, amelyikben ne lenne névelírás, helyesírási hiba, sorrendtévesztés, szerkezeti logikátlanság, következetlen írásmód stb.
3. Annyira nem tiszteljük a magyar nyelvet, hogy a kinyomtatott címkéken „technikai okokból" egyszerűen lemondunk a korrekt magyar ékezetek alkalmazásáról.

Amikor tehát támogatni szeretnénk új címtárprogramok írását és terjesztését, ezekre a fejlett nemzetközi normáktól eltérő sajátosságokra gondolunk. És még egy nem lényegtelen technikai körülményre, hogy Magyarországon számítógépes programból a telefonhívások lebonyolítása ma még szinte kivitelezhetetlen. (Nem kell részleteznem, hogy miért!) Hiába vannak tehát főnöki-titkári funkciók ellátására készült szuperprogramok vagy minden raffinériával felszerelt integrált csomagok, egyszerűbb, ha taxiba ülünk! Ezért javasoltuk azt, hogy első lépésben legyen a címtár inkább csak egy nagyon könnyen kezelhető, képernyőn gyorsan fellapozható és kinyomtatható, minden titkárnő gépében ott lapuló kisokos... hát így lett a neve Kisokos.

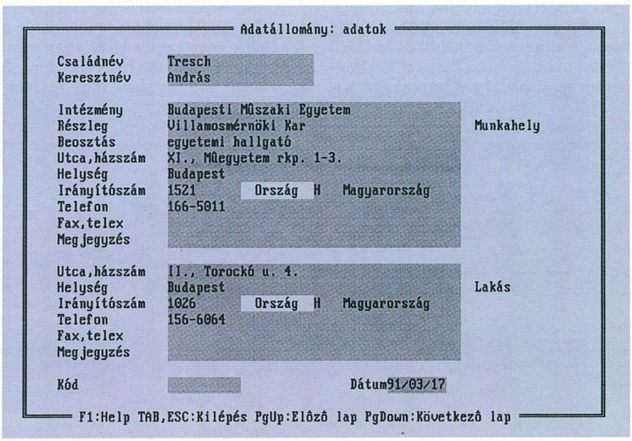
A címek kezelésében kialakult össznépi slendriánságot természetesen nem lehet sem egy programmal, sem más eszközzel rövid idő alatt megváltoztatni. De egy program is alkalmas lehet arra, hogy a rendezettség, a pontosság, a naprakész címnyilvántartás iránt egy kis szimpátiát ébresszen.

Már az vonzó, hogy a Kisokosban tetszőlegesen lehet az SZKI-Ventura vagy a CWI kódkiosztás szerint megjeleníteni és kinyomtatni a magyar ékezeteket. Kellemes játék, hogy mindenki saját ízlése szerint átdefiniálhatja a színeket és kikísérletezheti a neki legkellemesebb képernyőhatást. A program gyors kezeléséhez az egér használatának lehetősége is hozzájárul, bár a többség bizonyára megelégszik a billenytűzettel. Meghatározhatjuk, hogy a névsor a személyek vagy az intézmények neve alapján legyen a teljes magyar ábécé szerint sorbarendezve. Szintén magunk állíthatjuk be a kinyomtatandó címke típusát, magasságát és az oszlopok számát.

A programot tartalmazó könyvtárban a következő állományok szerepelnek:

kisokos.exe — A főprogram
kisokos.000 — Kisegítő állomány
kisokos.002 — Kisegítő állomány
kisokos.lll — Egy névsorformátum
adatok.cdb — Minta adatállomány
adatok.in1 — A minta 1. indexe
adatok.in2 — A minta 2. indexe

A program tetszőleges számú adat- és névsorformátum-állomány közül egyszerre mindig egyet használ. Az egy adatállományba felvitt adatlapok számát viszont csak a háttértároló kapacitása korlátozza. A mágneslemez mellékleten lévő demó változat néhány ponton persze eltér a komplett programtól, így például csak egy adatállományt kezel, és abban összesen 35 adatlap tölthető ki, az is csak akkor, ha előbb a programot (a fenti fájlokat) átmásoljuk egy másik lemezre, amelyiken van hely a terjeszkedésre. A demó verzióban értelemszerűen nem működik a másolási funkció és a segédinformációk csak CWI kódkiosztásban jelennek meg. Ahhoz mindenesetre a leszűkített demó változat bőven elegendő, hogy felmérjük a program „jóságfokát", és dönteni tudjunk megvásárlásáról.



Faklen Pál

Wampum varázslat?

A Wampum adatbázis-kezelő új verzióval gazdagodott. Sok mindent megtart a régiből, de van néhány új funkciója is. Ha hihetünk a szerző, Ward Mundy dokumentációjának, akkor keresve se találunk jobb programot az adatbankok piacán. A programot a DOS Shareware-ben megjelent cikk alapján ismertetjük.

A Wampum 4.1-es változata első látásra nem sok változást mutat. (Figyelem! A SolarSoft #51-es lemeze jelenleg még csak a 3.2b verziót tartalmazza!) A menük felépítése olyan maradt, amilyen volt, és nem is bővült semmivel. A döntő újdonságok inkább a rendszer mélyén vannak elrejtve. Ezekből akkor kapunk először ízelítőt, amikor kiválasztjuk a „file select" menüpontot. Kipattan belőle egy ablakocska a rendelkezésre álló fájlok neveivel, valamint a lehetőséggel, hogy új fájlt létesítsünk vagy esetleg az elérési utat változtassuk meg. A fájlok és lehetőségek között a kurzorvezérlő billentyűkkel mozgunk, amelyek a sorkiemelést fel-le mozgatják. A megerősítés természetesen az ENTER billentyűvel történik.

Mindez elvileg semmi különös, de jelentős begépelési munkától szabadítja meg a Wampum-használót, és az előző változatokhoz képest maga a kipárnázott luxus. Ezeket az ablakocskákat azután szinte mindegyik menüpontban megtaláljuk, az éppen érvényes követelmények szerint változó tartalommal és funkcióval. Nem lenne érdemes most minden funkción végigmenni, az öreg róka Wampum-felhasználók ezeket már kívülről tudják.

Bemutatjuk viszont a már említett új ALT billentyű-kombinációkat. Ezek a főmenüből hívhatók.

ALT-S

Ez a billentyű-kombináció meghívja a SELECT ASSIST funkciót. Ez nem újdonság ugyan a Wampumban, mégis érdemes néhány szót szólni róla. A képernyőn egy négy mezőre osztott beíró sor jelenik meg. Az első mezőben látható a pillanatnyilag aktív adatbank fájl neve. A fel-le billentyűk használatával kiválaszthatjuk azt az adatmezőt, amelyben keresni kívánunk. A kiválasztást az ENTER erősíti meg.

A második mezőben adhatjuk meg a keresési feltételeket. Itt Boole-algebrai fogalmak szerint válogathatunk, szintén a fel-le billentyűkkel. Most is ENTER-rel erősíthetjük meg a választást. A harmadik mezőbe írjuk be a keresett fogalmat. Itt már nincs mankó – be kell írnunk, amit keresünk és ENTER.

Az utolsó mezőben ismét az iránybillentyűkkel választunk, hogy akarjuk-e a feltételek összekapcsolását vagy sem. Ha nem, akkor üresen hagyjuk a mezőt, egyébként úgy járunk el, mint fent. Ha minden összekapcsolást elvégeztünk, akkor a definíciót a PgDn billentyűvel zárjuk le. Ez a funkció tehát a kevésbé járatos felhasználónak is megkönnyíti az adatok megtalálását.

ALT-I

Az „I" ebben az esetben INDEX-et jelent. Ezzel a funkcióval gyorsan újraindexelhetjük az aktív adatállományt. Itt is kipattan egy ablak, amelyben megtaláljuk az összes aktív indexfájlt. Az elsődleges indexet az ablak első, elkülönített mezejében találjuk. A kívánt új elsődleges indexet ismét az iránybillentyűkkel választhatjuk ki, és az ENTER-rel erősíthetjük meg. Ily módon másodpercek alatt át tudunk váltani az egyik indexből a másikba, anélkül, hogy a főmenün és indexmenün át kellene időrabló műveleteket végeznünk.

ALT-M

A Wampum felkínálja a lehetőséget használója számára, hogy különböző menüket készítsen. Ezek a programozó szándékának megfelelően az adatállományok feldolgozása során bizonyos funkciókat, illetve lehetőségeket nyitva hagynak, másokat viszont kizárnak. Az ALT-M kombináció lenyomásával megnyílik az az ablak, amelyben a több munkahelyes rendszereken dolgozók vagy az egy közös gépet használók közül néhánynak bizonyos dolgokat, pél-

ADATLAP

Név:	Wampum v.4.1
Szerző:	Ward Mundy, USA, 1990
Leírás:	dBase kompatibilis adatbázis-kezelő
Konfiguráció:	minimum DOS 2.1 és 512 k RAM, merevlemez ajánlott

dául az adatfájlok törlését vagy megváltoztatását megtilthatjuk.

ALT-B

Ez a billentyű-kombináció egy teljesen új Wampum-funkciót bocsát rendelkezésünkre. A szerző „power browse"-nak nevezi, a „turkálás" egyszerű és különlegesen hatékony módszerének. Ha kiválasztottuk a kívánt állományt, valamint az indexfájlokat és a keresési kritériumokat, ez az újdonság még a dörzsölt felhasználót is csodálkozásra készteti. Az adatok ugyanis táblázatos formában kerülnek a képernyőre. Minden egyes mező szépen bekeretezve. És ezzel elkezdődhet a kaland. A „power browse" ugyanis lehetővé teszi a felhasználónak, hogy új adatcsoportokat vigyen be, a meglévőket átszerkessze vagy törölje, és hogy bizonyos esetekben egy-egy ilyen lépést szükség esetén vissza is vonhasson.

Hadd írjuk le röviden, anélkül, hogy a részletekbe mennénk, miként dolgozhatunk ezzel a funkcióval. Az ALT-B hatására egy ablakocska jelenik meg a képernyőn, amely jelzi, hogy a „browse" üzemmódban vagyunk. Lehetőségünk van az összes vagy csak bizonyos kiválasztott adatmezőkkel dolgozni. Ha már megadtunk egy válogatási feltételt, azonnal megkezdődik az adatok kiírása a képernyőre. Mindig 15 adatcsoportot jelenít meg a program. A képernyő fennmaradó részét a segédablakok foglalják el.

Most el lehet kezdeni a munkát, egy bizonyos adatcsoport kiválasztását az iránybillentyűkkel. A szerkesztéshez a kurzort a szükséges adatmezőre léptetjük, és egyszerűen írni kezdünk. De azt is megtehetjük, hogy a mezőt előbb ENTER-rel megerősítjük, és csak ezután kezdünk a szerkesztéshez. Ha készen vagyunk, akkor az adatcsoportot ENTER-rel írjuk a fájlba.

Ugyanígy járunk el adatcsoportok kitörlése esetén. Amint fentebb leírtuk, kiválasztjuk a csoportot, és egyszerűen lenyomjuk a DEL billentyűt. Ha tévedésből nem a megfelelő adatcsoportot töröltük volna ki, ezen a ponton még érvényteleníthetjük a műveletet. Ha ugyanis ismételten megnyomjuk a DEL billentyűt, akkor a kitörölt adatcsoport ismét megjelenik.

Hasonló módon lehet új adatcsoportokat létesíteni. Kiválasztjuk a maszk sorrendben legutolsó adatcsoportját, és megnyomjuk a lefelé mutató iránybillentyűt. Ekkor egy hangjelzés és egy felirat közli, hogy az „add" üzemmódba léptünk. A felfelé mutató iránybillentyűvel visszatérhetünk az előző üzemmódba. Az adatok beadása gyerekjáték. Oda kell kormányozni a megfelelő mezőbe a kurzort, az adatokat beírni és ENTER-rel megerősíteni. Ennyi.

A további újításokat most nem tárgyaljuk, hiszen a Wampum dokumentációk elegendő felvilágosítással szolgálnak. De néhányat legalább felsorolunk. Így például a Wampumot tárrezidens programként is behívhatjuk, amihez maximum kilenc „look up"-fájlt rendelhetünk hozzá. Kellemes szolgáltatás továbbá az a kommunikációs ablak, amelyben egy Hayes-modem segítségével az adatbankunkban lévő telefonszámok bármelyikét közvetlenül felhívhatjuk.

A Wampum 4.1 az új funkciók révén munkát takarít meg, könnyebb vele dolgozni, mint az előző változatokkal. Egy-két hiányossága azonban még mindig maradt. Így például a „report generator" meghívása után nekem nem sikerült a Wampumba visszatérnem. A rendszer egyszerűen lefagyott. Egyébként sem tartom a Wampumot kezdők számára ajánlható programnak, ehhez a funkciói és lehetőségei túl komplexek. Mindenesetre a gyakorlott felhasználónak mindent nyújt, ami az adatbázis-kezeléshez szükséges. Még akkor is, ha ez az adatbázis-kezelő nem mindig és nem mindenben tökéletes.

Szekfű András

SolarSoft sikerlista

Az 1991. januári és februári eladások alapján

	No.	Programnév	Db	Programleírás
1.	421	PKZ110 & ZIPDMP & SHEZ	1	A „sűrítés" magasiskolája és Norton Commandere
2.	319	SCAN72 & OTHERS	1	McAfee-féle vírusmegelőző, -detektor és -ölő
3.	423	QFONT 1.15B	1	Szoftfonteditor magyar Venturához!!!
4.	096	AS-EASY-AS 4.00P	1	Lotus-kompatibilis táblázatkezelő, egyszerűbb
5.	435	OPTIKS & ICONVERT	1	PCX,PIC,GIF,TIF,GEM,MAC... grafikus konverterek
6.	432	LZEXE & LIST 7.5E	1	Gyors EXE kompresszor, Vernon Buerg LIST PLUS-a
7.	304	TURBO TECHNO JOCKS	2	Szuper Turbo Pascal unitok forrásaikkal!
8.	425	POP-DBF 1.1 & DLITE	1	Tárrezidens dBase (EDIT/BROWSE/DISP STRU/APPEND)
9.	442	WINDOW PRO 1.51	1	No.1 ablaktechnika C nyelven (small/medium/large LIB)
10.	427	ANADISK, CON-FMT	1	Lemez- fájl- és FAT-editor, rezidens formattáló!
11.	441	DATABASE IN C	1	Adatbázis-kezelés C-ben, forráskóddal (Btree, dBase)
12.	383	4DOS V3.01A	1	COMMAND.COM pótló DOS héj: 50 új parancs
13.	461	ZEPHYR 2.0	2	Komplett interaktív adatbázis-kezelő
14.	356	VGA GAMES #2	1	8 játék (Flipper, új Tetris-klón, rulett, légyfogó)
15.	440	CHESS	1	7 különféle sakkprogram + egy sakkóra
16.	437	EGA GAMES #1	1	3D pingpong, nehezített Tetris és amőba stb.
17.	329	PC-MAGAZINE BENCHMARK	1	Az USA szaklap hardvertesztjei szervizeseknek (v5.0)
18.	456	WP 5.X DRIVERS	1	Komplett Postscript fontdefiníciók
19.	070	BLACK MAGIC	3	Grafikus módú hipertext, önálló futtató modullal
20.	446	C-MIX #4	1	On-line változtatható méretű RAMDISK EMS/XMS-ben
21.	454	WP 5.X MENUS & MOUSE	1	Menükészítés makrókkal, egérrel WordPerfectben
22.	154	GETFILE & MAXIFORM	1	Formázás után DS/DD: 420 KB, DS/HD:1.4 MB!
23.	458	WP 5.X MACROS & WORDS	1	Helyesíráskorrektorok, szótárak, mintamakrók
24.	439	BRIDGE, CANASTA	1	Kártyajátékok: Pikk dáma, bridzs (BAS forrás!)
25.	333	TURBO SPRITES	1	Grafikus tervezés és animáció Turbo Pascalban
26.	457	WP 5.X TOOLS	1	Szövegformátum-átalakítók (WS, XyWrite, Word)
27.	445	C-MIX #3	1	3D grafika és C fordító C forráskóddal!
28.	453	WP 5.X INFOS & TUTOR	1	Makrózást oktató program WordPerfecthez
29.	384	HEXCALIBUR	1	RAM-editor blokkműveletekkel (insert/delete/move)
30.	459	WP 5.X GRAPHICS	1	45 darab, WordPerfectbe beemelhető ábra (WPG)

Vegyes shareware-hírhalom

A #319-es számú SCAN... lemez tartalmát ismét felfrissítették, immár a McAfee-féle SCAN és CLEAN 6.8B74 változata (SCAN74B, CLEAN74B) található a lemezen.

A #321-es PKZ110... lemez sem kerülte el az időszerű felújítást, itt Phil Katz PKLITE 1.03-as EXE és COM tömörítőjét adták közre a SHEZ 5.9 társaságában. A PKLITE 1.03 már hatékonyabban zsugorít, mint az LZEXE, így annak méltó vetélytársa lett. Várjuk az LZEXE francia programozójának csattanós válaszát...

Egy érdekes hír: a #418-as FAST/SOFA/FFD egyedi gépközeli programnyelv regisztrált változatát hosszas és állhatatos keresés és egy Ausztráliában járt számítástechnikus közbenjárásával sikerült beszereznie az Eötvös Klubban kedd esténként rendszeresen találkozó SoftWing csoport vezetőjének, Barabási Rezsőnek. A program egyenesen Új-Zélandból, a kivik szigetéről érkezett, sajnos egy sérült „noname" (nevesincs) lemezen. Számítógéppel már nem lehetett elolvasni a megtörött hordozófelületű floppyt. A Cédrus speciális floppymásológépe azonban jelesre vizsgázott, minden további nélkül hibátlan másolatot készített a már már kétségbeesett tulajdonos őszinte megelégedésére. Nem boszorkányság, technika. (Ez most nem reklám.)

Tud............. /19.............sz.

ladó (meghatalmazottja) felszólalt:

.........., 1991 .... hó ..... -ig

A feladónak az összeg rendeltetésére vonatkozó közleménye

A bankszerv teljesítését igazoló bélyegzőnyomat:

Trükkök a PC-n

Hangok a háttérben

Az IBM PC-n elterjedt programnyelvek közül szinte mindegyik támogatja a különböző magasságú hangok megszólaltatását, amelyet némi jóindulattal akár zenének is nevezhetünk. A sokféle géptípus azonban megkeseríti azoknak az életét, akik programjaikat zenével szeretnék élvezetesebbé – esetleg elviselhetetlenné – tenni.

Kínos meglepetés érheti azt a botcsinálta szerzőt, aki a gondosan elkészített dallamot egy másik gépen is szeretné meghallgatni. A PC-k változatos műveleti sebessége miatt a muzsika tempója akár a tízszeresére is növekedhet vagy csökkenhet. Szükséges tehát valamilyen időzítés alkalmazása, amely meggátolja az effajta sebességingadozásokat. Jó ötletnek tűnhet – csak kissé komplikált megoldás – a pontos idő lekérdezése minden egyes hang megszólalása előtt; ez azonban felemésztené a program futási idejének jelentős hányadát. Éppen ezért olyan megoldást kell keresni, amely jellegénél fogva megoldja majd az időzítés minden gondját. Ezt a szerepet töltheti be az 1Ch sorszámú megszakítás, amit éppen ilyen célokra találtak ki.

Az 1Ch – felhasználói megszakítás

Szinte mindenki találkozott már olyan programmal, amely a pontos időt jeleníti meg a képernyő sarkában. Ezek általában úgy működnek, hogy magukra irányítják az 1Ch megszakításvektort, és bent maradnak a tárban. Az említett vektor ekkor egy olyan kódrészletre mutat, amely időről időre végrehajtódik, s ennek eredménye az idő folyamatos kijelzése. A zene megszólalását is ütemezhetjük az óraprogramoktól ellesett módon. Ehhez olyan eljárást kell készíteni, amely egyrészt interrupt típusú, másrészt képes egy adott táblázat soron következő elemét beolvasni. Ezzel két legyet üthetünk egy csapásra: nemcsak az időzítés gondját oldjuk meg, hanem lehetővé válik a zene megszólaltatása is a háttérben, azaz vidám csárdással szórakoztathatjuk a felhasználót, miközben a címlista szortírozása folyik...

Az 1Ch megszakításra írt zene elkészítésénél feltétlenül figyelembe kell venni, hogy a rutin másodpercenként tizennyolcszor hajtódik végre, tehát a ritmus alapegysége ennél nem lehet rövidebb. Egyszólamú dallamok esetében ez kielégítő, de a polifóniához sajnos kevés. A lemezen mellékelt MUSIC.EXE illetve MUSIC.PAS programban látható egy félmegoldás az egynél több szólamú zene lejátszásához, ám mindjárt az első hangok megszólalása után rá kell jönnünk, hogy ez nem az igazi. Valódi polifónia létrehozása a PC hangszóróján igen fáradságos, s nem csak a kis méretek miatt.

A PC beépített hangszórójánál ugyanis már csak az azt meghajtó elektronika primitívebb; lényegében csak négyszögjelek generálására alkalmas. Magyarán szólva: berreg. Léteznek persze olyan programok, amelyek ennek ellenére beszélnek vagy digitalizált zenét játszanak, de ezek egyrészt közvetlenül vezérlik a hangszórót – aminek mikéntjéről még lesz szó –, másrészt hardver szükséges az „alapanyag" rögzítéséhez. (A kereskedelemben kapható olyan szoftver, amely kártyatervet – NYÁK-ot – is ad a digitalizáláshoz.) Némi trükkel azonban mi is javíthatunk helyzetünkön: átállíthatjuk például az időzítő áramkört magasabb fordulatszámra, s így nem köt többé bennünket az egytizennyolcados alsó határ.

A timer felpörgetése

A számítógép belső hatáskörébe tartozó hardvert önkényesen átállítani a legnagyobb udvariatlanságok egyike, de esetenként szemet hunyhatunk az effajta ügyködések felett. Különösen akkor, ha az eredmény másként nem vagy csak nehezen érhető el. Ilyen eset áll elő akkor, ha olyan zajt óhajtunk csapni, amelyhez gyors futású ciklusműveletek szükségesek. Egy lassabb gépen a gondosan kikísérletezett effekt csapnivaló prüntyörgéssé silányul, ha nem tudjuk valamilyen biztos módszerrel garantálni a tempó állandóságát.

Ilyenkor az 1Ch megszakítás túlságosan ritka járatát kissé felpörgetve megfelelő időzítést lehet elérni. Ehhez a 8253-as timer áramkört kell beállítani a következő módon: a 43h porton keresztül üzenhetünk a timernek, itt választhatjuk ki a kívánt csatornát. A kiküldött bájt két legmagasabb helyiértékű bitje tartalmazza a csatorna számát, amely esetünkben a nullás. A negyedik és az ötödik bit az adat stílusát mutatja; itt két egymás utáni bájtot fogunk küldeni, tehát bináris 11 az eredmény. A következő 3 bit az üzemmódváltó, amely egyébként a lehetséges 8 közül csak hatféle lehet, – nekünk a hármas (011) kell. A legkisebb helyiértékű bittel választhatunk a BCD vagy a bináris számformátum között (nulla egyenlő bináris). Ha jól figyeltünk, akkor a számokat összerakva magkapjuk a 00110110b értéket, ez kerül tehát a 43h-as portra.

Ezek után jön az osztó kivitele a 40h sorszámú portra. A 8253 bemenetén kapott standard 1 193 181 Hz, azaz durván 1,2 MHz-es jelet osztja a 40h-es porton található szavas érték, amely alapesetben 0, pontosabban 65 536. Ha vesszük a fáradságot és elvégezzük az osztási műveletet, az eredmény 18,2 , ami azt jelenti, hogy másodpercenként tizennyolc jelet kapunk az áramkör kimenetén. Ez az érték határozza meg azt is, hogy az 1Ch megszakításra írt szubrutin milyen gyakran fusson le. Ha tehát a 40h-es portra kiírt adattal elosztjuk a bemeneti frekvenciát, megkapjuk az 1Ch megszakítás ismétlési frekvenciáját.

A lemezmellékleten található programban – EFFECT.PAS, EFFECT .EXE – a kiküldött érték 1000; ez mintegy hússzorosára gyorsítja az említett interruptot. Így másodpercenként kb. százszor fut le az a kód, ami normális esetben csak tizennyolcszor.

A módszernek sajnos jócskán akadnak hátrányai is, épp ezért nagy gondosságot kíván attól, aki arra vetemedik, hogy saját programjában kihasználja a timer programozásának sajátosságait. Az egyik fontos tény, hogy a számítógép belső órája is sietni fog, ha átállítjuk a kimeneti frekvenciát. Ez XT-k esetében nem olyan komoly dolog, de elemes órával ellátott gépeknél igazán illetlenség összezagyválni az időt. Épp ezért a timer programozása után feltétlenül be kell építeni valamely alkalomhoz illő időmérő rutint, amely a futás befejeztével intelligensen visszaállítja az akár napokkal is elcsúszott időt. Esetleg az AT belső órájából is kiolvashatjuk az eredeti értéket.

Végezetül néhány apró tanáccsal szeretném elrettenteni az esetleg nekibátorodott olvasókat. Senki ne lepődjön meg azon, ha a lemezműveletek sorra kudarcba fulladnak a timer felpörgetése után. Ilyenkor célszerű visszaállni a megszokott értékre (0), és máris folyhat az adatok áramlása. A multitask rendszerek sem igazán lelkesednek azért, hogy átállítgatjuk az időzítéseket, s többnyire lefagyott programmal jutalmazzák az ilyen ténykedéseket.

Boros György

Irodalom
Boér – Dóra: Az IBM PC-k belső felépítése – LSI, Budapest, 1989.

Shareware-országi utazások

A keresztapa

Bob Wallace a shareware egyik úttörője. Bár PC-Write programját az amerikai szaksajtóban évek óta üdvözlik lelkesült híradásokkal, azt csak kevesen tudják róla, hogy ő ennek az üzletágnak a névadója, hogy ő a keresztapa. A PC-Sig Magazine 1989. novemberi száma terjedelmes interjúban mutatta be őt. Az alábbiakban ennek rövidített változatát közöljük.

– Valóban nem Ön találta ki a „shareware" szót?

BW: Ezt a kifejezést mint rovatcímet láttam, és megkérdeztem az íróját, alkalmazhatom-e a felhasználók által támogatott szoftverek jelölésére. A „felhasználó által támogatott" nagyon körülményes kifejezés, és egy rövidebb szó biztosan győz. A shareware rövid, szinte fülbemászó meghatározás. A rovat megszűnt, és a szerző kérésemhez hozzájárult.

– Régóta otthon van a shareware piacon?

BW: Igen. 1983 februárjában elhatároztam, hogy készítek egy szövegszerkesztő programot és shareware-ként piacra viszem; ezért kiléptem a Microsofttól és megalapítottam saját vállalatomat, a Quicksoftot. Azért döntöttem a szövegszerkesztés mellett, mert egy olyan, sokak által használható programot akartam készíteni, ami elősegíti a felhasználók kreativitását. Az írott szó az, ami a tudományos és kulturális fejlődést civilizációvá emeli. Ha tehát segíthetek, hogy mindenki bővíthesse írásbeli tudását, a civilizációt fejlesztem.

– Így tehát ez személyes ambíciója volt?

BW: Nos, szeretem a szövegszerkesztést, és azt gondoltam, jó lenne, ha mások is megszeretnék. Arra is gondoltam, hogy egy szövegszerkesztő programot megvásárlása előtt mindenki szeretné ki is próbálni.

– Fontos, hogy a felhasználók elégedettek legyenek a szövegszerkesztő programmal?

BW: Az emberek olyan alkotóképes szoftvereljárásban érdekeltek, mely ugyanúgy gondolkodik és cselekszik, mint ők. Ez nem egészen így van az adatbázis-kezelők és a táblázatkezelők esetében. Egy vállalat alkalmazottai kénytelenek egységes adatbázis-kezelőt használni, mert osztoznak az adatokon. De a vállalatok megengedhetik maguknak, hogy alkalmazottaik megválasszák saját szövegszerkesztő programjaikat mindaddig, amíg azokat ésszerűen tudják oda-vissza konvertálni. Nem olyan fontos, hogy mindenki ugyanazzal a szövegszerkesztővel dolgozzon. Különböznek a vélemények arról, hogy melyik a legjobb szövegszerkesztő program, de általánosan egyik sem elfogadott.

– A szövegszerkesztő programot egyéni igényekhez igazította?

BW: Amikor először gondoltam arra, hogy szövegszerkesztő programot készítek, a WordStar volt a standard program, de a legtöbben ezt nem kedvelték. Gondoltam, talán tudok jobbat készíteni. A PC volt az első számítógép, amely funkcióbillentyűkkel és nagy számú egyéb billentyűvel rendelkezett; és a PC-Write nagyon is billentyűzet-orientált.

A szövegszerkesztő programok egy része vizuális beállítottságú embereknek készül: ezek képi ábrázolást és egeret használnak, mint például a Macintosh. De vannak mások, amelyeket az élő szóra és a nyelvészeti elemekre alapoznak, mint az XyWrite, amely szóbeli utasításokat alkalmaz. A vizuális emberek azt mondják: „látom, mire akarsz kilyukadni". Akik inkább verbálisak, azt gondolják: „hallom, mit akarsz mondani". A PC-Write kézzelfoghatóbb; felhasználója leginkább arra fog gondolni: „érzem, mit akarsz ezzel". A szövegszerkesztővel való kapcsolat számomra nagyon fontos, és ennek megfelelően szerkesztettem meg a PC-Write-ot.

– Merre tart a PC-Write?

BW: Egyre terjedelmesebb és jobb lesz. A 3.1 verziónak a kötőjel a jellegzetessége, és a kibővített memóriára is gondoltunk. Kiegészítő programok hozzáadására is lesz mód, mint például az Inset és a Wordfinder, ezek szorosabban kapcsolódnak majd a PC-Write-hoz. A Lite a PC-Write egyszerűbb változata. Ez azoknak készült, akik nem akarnak keresztülvergődni egy kiterjedt programon, hogy elvégezhessék feladatukat. Felhasználói közé tartozhatnak azok az írók és újságírók, akik nem igénylik a fejlettebb formattálást; az egyetemi hallgatók, akik tanulmányokat írnak; programozók, akik igénylik a szövegszerkesztést, és azok, akik például levelet akarnak írni és nincs szükségük a teljes, megformázott végeredményre, hanem csak annak tartalmára.

– Gondolt-e arra, hogy bevonuljon a desktop publishing birodalmába?

BW: Már elkezdtünk egy fejlesztést, amely a grafikákat foglalja magába; ez alapvetően desktop publishing lesz.

– Rendelkeznek-e különböző betűkészletekkel és betűméretekkel?

BW: Igen. Már van egy nagyon jó laserjet alátámasztásunk és egy bámulatos betűkészlet-válogatónk. A PC-Write kiszabja minden egyes karakter szélességét, és a bekezdést a karakterek tényleges szélességéhez mérten formázza meg.

– Saját maga írta a PC-Write programot?

BW: Az eredeti programot igen. A 3.0-ás verzió már egy hibrid. A szerkesztő részt én írtam, és valaki más a nyomtató részt. Nem tölthettem egész időmet a programmal, mert a társaság elnöki tisztét is el kellett látnom, és nem fordíthattam arra sem elég időt, hogy elnök legyek, mert szorgosan programoztam. Persze magánéletem sem nagyon volt. Megtanultam, hogy amikor egy vállalat meghatározott nagyságot elér, és én vagyok a megalapítója és egyben programozója is, el kell döntenem, hogy én legyek a vezető programozó és alkalmazzak egy elnököt, vagy vállaljam az elnökséget és szerezzek egy jó programozót. Én az utóbbit választottam. Így hát a PC-Write 3.2 verzióját Steve Levy fogja elkészíteni, és folyamatban van egy másik programozó szerződtetése is.

– Milyen más tulajdonságokat kíván még beépíteni a szövegszerkesztőbe?

BW: Az egyik funkció, amelyet a PC-Write 3.1 verzióba beépítettünk, a területkijelölés. Egy ilyen derékszögű területet később egy kulcsszóval, egy címkével találhatunk meg. A program a területen belül valamennyi számot összegezi számunkra. Beépítettünk egy automatikus sorszámozót is a sorok, bekezdések és lábjegyzetek számára.

– És mit tud mondani azokról a különleges szövegszerkesztőkről, amelye-

ket ügyvédek, vagy más, egyedi iratformátumot igénylők használhatnak?

BW: A specializált szövegszerkesztő jó ötlet, de nekünk minden erőnket annak kell szentelnünk, hogy egy programot fejlesszünk – jól.

– A shareware, mint üzleti technika, hogyan vált be az Ön számára?

BW: Igen jól. Első termékünket 1983 augusztusában bocsátottuk ki, és elküldtük példányait a sajtónak, a felhasználói csoportoknak és a nagyobb számítógép-forgalmazóknak. Miután pozitív válaszokat kaptunk, a cég elindult.

Ennél a pontnál a shareware valóban nem volt még piaci metódus. Elsősorban azok használták, akik teljes munkaidejük mellett szabadidejükben is dolgoztak egy-egy programon. Megengedték, hogy ezt mások is használják, és azután hozzájárulást kértek ezért. Ebben a helyzetben volt a PC-File és a PC-Talk szerzője. Számos shareware szerző a mai napig is ennél a pontnál tart.

Amikor a Quicksoft céggel elindultam, és elhatároztam, hogy a shareware-t mint marketing módszert fogom alkalmazni, kissé el kellett térnem az addigi gyakorlattól. Ahelyett, hogy azt mondtam volna: „Hozzájárulást szeretnénk kérni a szoftver használatáért. 25 dolláros adományra gondoltam.", egészen másként fogalmaztam. Azt mondtam: „Ez egy kétlépcsős eljárás. Az első lépésben szabadon kipróbálható. A másodikban ténylegesen megszerzi a szoftvert." Ezt a második lépést neveztem el „regisztrálás"-nak vagy „egy regisztrált példány megszerzésé"-nek.

Az első képernyőn tudattam a felhasználókkal, hogy ha pénzt küldenek nekünk, elküldjük számukra a kézikönyvet és a kereskedelmi forgalomban lévő szoftver legfrissebb verzióját; vállaljuk a rendszeres információküldést, közöljük a korszerűsítéseket, és egy évig biztosítjuk a technikai támogatást. Érzékelni fogják, hogy milyen kézzelfogható előnyöket jelent mindez. Természetesen számítottam az emberek üzleti tisztességére és kollektív hajlamára is – hiszen a regisztrált felhasználók köre végül is egyfajta közösség –, és az ebből fakadó rejtett előnyökre. De azt gondoltam, hogy ugyanakkor kézzelfoghatóbb dolgokat is kapniuk kell.

– Úgy érzi, hogy jobb helyzetben van ma, mintha egyáltalán nem foglalkozott volna a shareware megközelítéssel?

BW: Határozottan. Hogy ezt jobban megértsük, tegyük fel a kérdést: „Hányan fizették volna meg a teljes árat, ha ez nem shareware?" Ebbe mindenki beleértendő, aki használja a PC-Write programot, miután kapott egy példányt a barátjától, egy eladótól, egy felhasználói csoporttól, vagy bemutató példányként tőlünk. A válasz: sehányan, mert nem tudtam volna hirdetni. A hirdetés pedig nem is mindenkor eredményes. Most mi is hirdetünk, de elsődleges marketing eszközünk továbbra is a shareware.

Más kérdés, hogy hányan fizetnek magáért a shareware programért, a kereskedelmi változat igénye nélkül? Azt gondolom, hogy viszonylag kevesen. De akik nem fizetnek a shareware-ért, általában nem fizetnek a kereskedelmi szoftverekért sem. Ők lemásolják maguknak a PC-Write-ot, a WordPerfectet, és nem törődnek semmivel. Azért nem veszítünk, ha nem regisztráltatják magukat, mert egyébként sem fizetnének és nem regisztráltatnák magukat.

– A shareware szerzők többsége inkább a kiskereskedelmi piacon érdekelt. Mi erről a véleménye?

BW: Jelenleg üzleti forgalmunk 10 százaléka viszonteladók útján bonyolódik le. Most a dobozrajzolón dolgozunk, de ha ezzel elkészülünk, tárgyalást kezdeményezünk az Eggheaddel. A kiskereskedelembe való belépés komoly tervezést és sok pénzt igényel. Bizonyos feladatokat az áruházakkal közvetlenül akarunk lebonyolítani, lemezeket küldünk nekik, tárgyalunk velük, és egy kicsit gyakoroljuk is magunkat. A kiskereskedelemben való részvétel hasznos.

– Mennyire érdekes Ön számára a külföldi piac?

BW: Nagyon érdekes. Programjaink külföldön nem shareware-ek. Nagyon jól állunk Európában. A PC-Write-ot Franciaországban, Angliában, Hollandiában és az NSZK-ban adjuk el.

A PC-Write Ausztráliában a Safesoft útján került eladásra, de ez nem volt eredményes – a terjesztésért túl sokat számítottak fel. Azt tervezzük, hogy Ausztráliában shareware lesz a PC-Write; a mi kézikönyvünket fogjuk használni, az angol helyesírás-ellenőrzéssel és a helyi technikai támogatással. Ez egy kísérlet. Ha a shareware működik Ausztráliában, működhet a Szovjetunióban is. Ámbár a Szovjetunióban a szoftver-csatornák nagyon primitívek.

– Mit tanácsol azoknak, akik szoftver termékület shareware-ként szándékoznak eladni?

BW: Ha valaki egy vállalkozást akar elindítani, a shareware nagyon jó metódus a szoftverértékesítés számára. De a szoftver értékének többnek kell lennie, mint a puszta termék. Érdemes mérlegelni, hogy kézikönyvet, technikai támogatást és más extrákat ajánljunk. Ha adóprogramunk van, akkor információkat kínálhatunk azoknak, akik regisztráltatták magukat.

Azok ellen a szerzők ellen vagyok, akik egyes funkciókat letiltanak shareware programjukban. A vásárlók ezt nem értékelik, és elfogultak lesznek a szerzőkkel szemben, hiszen nem tudják kipróbálni a teljes szoftvert. A mi shareware-verziónk azonos a regisztrálttal.

– Vannak programok, melyek nem jól működnek shareware-ként?

BW: Hogyne. Ilyenek a játékok, ezek hajlamosak arra, hogy „elfogyasszák" őket. Ha egyszer már kiókumláltuk, hogyan kell megölni a sárkányt, akkor végeztünk is velük. Ugyanez igaz az oktatóprogramokra is. Amint megtanultuk őket, többé már nem kellenek.

– Csak a fontosabb programok igényelnek támogatást és válnak shareware-jelöltté?

BW: Nehéz ezt állítani. Jó ellenpélda az Automenu. Nem igényel sok támogatást, eléggé egyszerű, kezdőknek való. De úgy gondolom, többnyire a nagyobb csomagnak magasabb a fuvardíja. Nagyobb és bonyolultabb csomaghoz a felhasználónak több dokumentációra és technikai támogatásra van szüksége, és a szerzők méltányos ellenértéket kérhetnek.

– Van még valami, amit hozzátenne, amiről azt gondolja, hogy érdekli a shareware közösséget?

BW: Még egy dolog van. A shareware egy kereskedelmi marketingtechnika, de egyben filozófia is. Aki nagyon megrökönyödik azon, hogy szoftverét használják és egy fillért sem fizetnek érte, az nem való a shareware-üzletbe. De ha valóban azt szeretné, hogy rengeteg ember használja a szoftverét, és ezért tisztességes pénzhez jusson, akkor ez igazi shareware-érzés.

Azoknak a százalékos aránya, akik regisztráltatják magukat, nem jelentős szám. Ha elegendő regisztrációs pénzt kap, és ebből megfizetheti alkalmazottait, lemezeket vásárolhat, és még csinos haszna is marad – nos, ez az a szám, ami érdekes. Tekintse potenciális vevőnek azokat, akik nem regisztráltatják magukat, és törődjön azokkal, akik ezt megteszik. Legyen biztos abban, hogy ügyfelei boldogok: jó dokumentáció, szoftver, technikai támogatás és napi információ áll rendelkezésükre. Szállítson a regisztrálást igénylőknek minőségi terméket. Ha értük cselekszik, akkor azok is felkeresik, akik eddig nem tették. Ez ugye egyszerű.

(Fordította: Ábrányi Zoltán)

Vonalkódok PC-vel

Ma már az Egyesült Államokban és Nyugat-Európában csaknem minden kereskedelmi forgalomban levő árun megtaláljuk az azonosító vonalkódot. Az amerikai Bartender igen speciális shareware program: lehetővé teszi, hogy vonalkódokat állítsunk elő gépünk párhuzamos csatlakozójára kapcsolt nyomtatón.

A Bartender kiszámítja és kinyomtatja a különböző vonalsorozatokat. Ezek változó szélességű vonalakból és vonalközökből állnak. A program használata csak abban az esetben tűnik ésszerűnek, ha a PC-hez vonalkódolvasó is csatlakozik, és ez az alábbi vonalkódtípusok valamelyikével dolgozik:

Barcodes, UPC-A, UPC-E, EAN/JAN-12, EAN/JAN-8, 3 of 9, Extended 3 of 9, Interleaved 2 of 5, Code 128, Codebar, ZIP +4.

Természetesen értelmetlen lenne a vonalkód alkalmazása akkor, ha nem rendelkeznénk a különböző kódtípusok felismerési lehetőségével. A program a fenti tízféle vonalkódtípust kezeli. Mivel Amerikában fejlesztették ki, ezek az amerikai vonalkódszabványnak felelnek meg. A Bartender európai alkalmazása csakis akkor lehetséges, ha a cégen belüli vonalkódok az amerikai típusok valamelyikének megfeleltethetők.

Lézerrel az igazi

A program kezelése viszonylag egyszerű. Az egyes vonalkód-formátumokhoz különböző meghajtók tartoznak, ezek memóriarezidensként tölthetők be. Felhasználói programunkban kialakíthatunk olyan jeleket, amelyek a jövőben vonalkódok lesznek. Ezeket adott jelsorozat kiadásával tehetjük a rezidens meghajtók számára felismerhetővé. A két speciális jel közötti betűket, illetve számokat a kiválasztott vonalkódtípusnak megfelelően fordítja vonalakká.

Azt a jelsorozatot, amely a rezidens Bartender meghajtót aktivizálja, szabadon választhatjuk meg egy konfigurációs programmal. Ugyanezzel a konfigurációs programmal további paramétereket is beállíthatunk, például a vonalkód magasságát, a bal oldali papírszéltől való távolságot vagy a soronkénti nyomtatások számát tűs nyomtatók esetén. Az utóbbi funkció a 9 tűs nyomtatókat is alkalmassá teszi jó vonalkódok nyomtatására, ennek ellenére ajánlatos inkább lézernyomtatót használni, ha professzionális alkalmazásról van szó.

Szabadon konfigurálható

Ezek a paraméterek parancssori paraméterként is átadhatók az egyes Bartender meghajtóknak, mégpedig közvetlenül a vonalkódnak megfelelő nyomtatómeghajtó aktivizálásával. Az egyes vonalkódokhoz tartozó meghajtók mindegyike 30 kilobájt memóriát foglal. Nagyobb meghajtó lehetővé teszi, hogy a konfigurációs programmal vonalkódtípust válasszunk, így az összes vonalkód egyidejűleg rendelkezésre áll a memóriában.

Nagyon jó, hogy a rezidens meghajtóval és a definiálható jelkészlettel tetszőleges programból elő tudunk állítani vonalkódokat. Ezáltal válik lehetővé szövegfeldolgozáskor a vonalkód és a szöveg keverése, sőt, árukészletet kezelő adatbázissal is kényelmesen előállíthatunk, illetve létrehozhatunk vonalkódokat. Programutasítással nyomtatás közben változtathatunk paramétereket, például a vonalkód magasságát, így a kódot az adott termékhez igazíthatjuk, anélkül, hogy a konfigurációs programot be kellene hívnunk.

Értékelés

A Bartender program használata olyan cégeknél ajánlott, amelyek vonalkóddal akarnak dolgozni, de még nem alakították ki az amerikai szabványokkal ütköző belső kódrendszert. Kényelmesen előállíthatunk programból is vonalkódot, a kódokat rezidens nyomtatómeghajtó generálja és nyomtatja ki. Már a tűs nyomtatókkal is kielégítő eredményeket kapunk. Igazán hatékony alkalmazásához azonban érdemes a PC-re vonalkódolvasót is csatlakoztatni.

Verebély Pálné

Jön, jön, jön!

A SolarSoft shareware-primőrjei

Egy speciális, igen jól használható, nyomtatott áramkör (NYÁK) tervező program jelenik meg a magyar shareware szekcióban. Az amerikai eredetű, kétlemezes, #043-as számú DROEGE programmal is méltán felveszi a versenyt képességeiben és szolgáltatásainak bőségében.

Szintén magyar készítésű a – ki hinné – BÉR nevű bérprogram, melyet egy tatabányai ügyviteli programokat készítő szoftverteam készített azzal a nem titkolt céllal, hogy a felhasználók ezen a teljes értékű programon keresztül megismervén őket, kedvet kapjanak más, kereskedelmi alapokon forgalmazott termékeik megvásárlásához.

Az M030-as számú CAB (computer aided banking) lemez kész Lotus 1-2-3-kompatibilis WK1 állományokat tartalmaz magyar nyelvű ismertetőkkel. A több mint 16 adattáblázatot tartalmazó válogatás „A bankár" című kéthetente megjelenő, elsősorban pénzügyesek, „tőzsdespekulánsok" olvasótáborát megcélzó mágneslemez magazin „Pénzügyi matematika" nevű állandó rovatából lett összeállítva. Jövedelemadó kiszámítása, kamatszámítás, részvények, kötvények matematikája, lízingszámítás és több lízingajánlat összevetésének megkönnyítése, devizatőzsdei aritmetika, ezek a fontosabb alkalmazási területek. Az érintett témák köre folyamatosan nő, a tervek szerint kéthavonta felújítják a CAB lemez tartalmát.

Az M029-es számú lemezen Turbo Pascal 5.5 fordítóhoz találunk tárgyorientált adatbáziskezelő mintaprogramokat és alkalmazást.

A SolarSoft érdekes kezdeményezésének ígérkezik az angol nyelvű biblia megjelentetése különböző lekérdező programokkal egyetemben. Bennfenteseknek kis prédikációkészítő (bibliai kereszttreferencia) segédprogramot is mellékeltek.

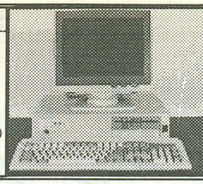
Sokak kérésére összeállításra került egy (esetleg később több) SolarSoft-Mix nevű válogatás, amely olyan sokak által keresett segédprogramokat foglal magába, mint a matematikai társprocesszor szoftverrel történő emulálása (a x87-es koprocesszort igénylő programok többsége ilyen chip hiányában nem indul, s ez a kis memóriarezidens program általában megoldja a problémát), a Borland VGA256.BGI és annak leírása, egyszerre több különféle AUTOEXEC.BAT és CONFIG.SYS állományt kezelő program stb.

hj

KOGINFORM
COMPUTER Kft.

AT286-16/VGA/40M

80286 CPU, 16 MHz órajel, 1M RAM
20MHz Landmark speed, NEAT alaplap
baby kivitel
Soros, Párhuzamos port
1.2M floppy
101 gombos billentyűzet
14" VGA monitor + VGA kártya (256K)
40M HD
1 év garancia



Ára: 117 500 Ft +ÁFA

	286-12	286-16	286-20	386SX	386-25	386-25C	386-33C	486-25C
CPU:	80286	80286	80286	80386SX	80386	80386	80386	80486
Órajel:	12 MHz	16 MHz	20 MHz	16 MHz	25 MHz	25 MHz	33 MHz	25 MHz
Landmark speed:	16	20	24	21	31	41	58	114
Cache memória	-	-	-	-	-	32K	64K	128K
Kivitel:	BABY			MINI TORONY		NAGY TORONY		
Alaprendszer: 1 Mbyte RAM/Soros port/Game port/Párhuzamos port/1.2 Mbyte floppy/101 gombos billentyűzet/1 év garancia								
Ár:(Ft)	AT286-12	AT286-16	AT286-20	AT386-SX	AT386-25	AT386-25C	AT386-33C	AT486-25C
Alaprendszer	43,200	44,700	53,500	74,300	97,100	122,200	125,800	240,200
2M RAM-al	50,400	51,900	60,700	81,500	104,300	130,400	133,000	247,400
4M RAM-al	64,800	66,300	75,100	95,900	118,700	143,800	147,400	261,800
8M RAM-al	-	-	103,900	124,700	147,500	172,600	176,200	290,600
Coprocessor	+16,000	16,000	-	+49,200	+67,700	+67,700	+73,800	-
Baby AT ház	0	0	0	-3,000	-3,000	-8,400	-8,400	-8,400
Minitorony kivitelben	+3,000	+3,000	+3,000	0	0	-5,400	-5,400	-5,400
Nagytorony kivitelben	+8,400	+8,400	+8,400	+5,400	+5,400	0	0	0

MONITOROK	
14" monochrom monitor+kártya	13,500
14" EGA monitor+kártya	39,000
14" VGA (800x600) monitor+kártya/256K	44,100
14" VGA (1024x768) monitor+kártya/512K	47,300
14" VGA (MULTISYNC) monitor+kártya/512K	60,400
19" VGA (1024x768) monitor+kártya	166,600
14" A/4 full-page (768x1024) monitor+kártya	53,900

Floppy/Hard Disk Drive	
1.2M vagy 1.44M floppy drive	7,100
40M HD 3,5"/28ms/AT-bus (Seagate)	28,700
80M HD 3,5"/17ms/AT-bus (Seagate)	53,400
160M HD 5,25"/17ms/SCSI (Maxtor) + kártya	170,100

KIEGÉSZÍTŐ EGYSÉGEK	
CAT billentyűzet beépített mouse-al	6,300
bus MOUSE	3,400
UPS 300 szünetmentes tápegység 300 W	34,800
Arcnet kártya (8bit/STAR)	4,600
Ethernet kártya (16 bit)	12,600
8 pólusú aktív HUB	14,800
MOBILE RACK cserélhető winchester fiók	6,400
serial MOUSE	2,300

Áraink a 25% ÁFÁ-t, valamint a helyszini installáció költségét nem tartalmazzák.

NAGY TÉTELBEN VISZONTELADÓKNAK KEDVEZMÉNYT ADUNK!

KOGINFORM—COMPUTER Kft. **1042 Budapest Tito u. 10.**

Telefon: (36-1) 1695146 Telefax: 1695146, 1604209 **Levélcím: 1327 Újpest 3. Pf. 44.**

Aki nem hiszi, menjen elébe!

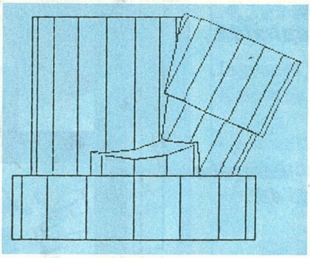
Ez a TRIOLA másképp szól

E cikk tárgyát egy olyan fantázianevű rendszer képezi, melyről elöljáróban kijelenthetjük, hogy mindenesetre a névválasztása találó.

A zenészeknek a triola három, általában egyenlő hangjegyből álló csoportot jelent, amelyet két hasonló, de ritmikai értékében normál hang megszólalásának időtartama alatt kell lejátszani. A CAD-ben jártas szakembereknek a TRIOLA azt a rendszert jelenti, amely az alig lábra kapó kétdimenziós korszakban három dimenzióban próbálta előadni a tervezőknek a testmodellezés alkalmazására irányuló halk futamait. És mindezt hazai földön, a SZÁMALK-ban.

Néhány, a testmodellezés partitúráját becsukott szemmel is lejátszó billentyűs és egy megszállott karmester alig három év alatt olyat produkált, ami külföldön is bekerült a repertoárba. Itthon pedig alig akadt vállalat, amely megérezte a tonalitást. Hozzátartozik az igazsághoz, hogy a legutóbbi időben a fejlesztői kamarazenekar inkább a hangerő növelésére koncentrált, mint az összhangzás feljavítására. Ugyanakkor nem a zeneművek hibája, ha a hallgatók nem ismerik fel a hangokban rejlő értékeiket. Vagy talán mégis igaz, hogy senki sem lehet próféta a saját hazájában?

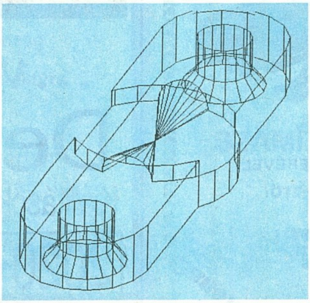
A TRIOLA-B egy tipikus, a CSG eljárást alkalmazó testmodellező rendszer. Olyan, mintha a fejlesztői kitűnően megtanulták volna egy tankönyvet, mely az elemi testek kombinálásának módszerét a mélységekig ismerteti. Vagy: mintha a tankönyv alapja lett volna a TRIOLA? Akár így, akár úgy nézzük, maga a rendszer a legbonyolultabb geometriai szituációkat, például az érintkezéseket, a többhéjú testek képződését, a metszetelésnél széteső objektumok keletkezését hibátlanul kezeli. Elsődlegesen a gépészeti objektumok (alkatrészek és összeállítások) modellezésére javasolható, de alkalmazásának nincs kizáró oka más szakterületeken sem. A rendszer mentes a külsőségektől, elsődlegesen a funkcionalitása dominál, és nem jellemző rá a hatásvadászat. A felhasználó szempontjából barátságos, könnyen áttekinthető, ami leginkább célirányos fejlesztésének eredménye. Ki lehet próbálni, demonstrációs változata hajlékony mágneslemezen



könnyen megszerezhető — a fejlesztőktől !!!

A „notáció" lehetőségei

A tervezési folyamatra vonatkozó információkat a rendszer a parancsok sorozataként őrzi meg. A modell igényelt módosításai az érintett parancsok megváltoztatásával egyszerűen végrehajthatók. Természetesen az eredmény újraszámítása időt vesz igénybe. A parancssorozat megfelelő részeinek többszöri feldolgozásával a variatív tervezéshez szükséges kiindulási objektumok prototípusai hozhatók létre. Az objektumok adatait a rendszer az élszomszédsági relációk alapján képezi le tárolási adatstruktúrává. A görbült felületű objektumokat a rendszer síklapokkal közelíti. A másodrendű felületek esetében a kanonikus alak paramétereit ugyancsak tárolja, így az NC megmunkáláshoz vagy a végeselemes analízishez kielégítő pontossággal számíthatók az áthatási görbék. A tárolási struktúra a csomópontok, élek és lapok adatai mellett tartalmazza a héj(ak)ra vonatkozó információkat is, ezáltal akár többhéjú objektumokon is végezhetők műveletek.

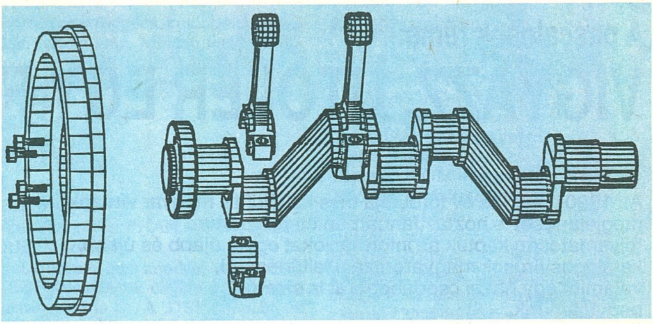


A TRIOLA rendszer elemi testjei a téglatest, az általános hasáb, az egyenes és ferde henger, a hengercikk, az egyenes és ferde kúp, a kúpcikk, a gömb és gömbszelet, a gömbsüveg és a tórusz. Szabálytalan 2D-s alapsíkidomból pásztázással vagy 2D-s kontúrgörbékből forgatással képezhetők testek. A síkidomok és a kontúrgörbék 2D-s rajzolórendszerekből importálhatók. Az elemi testeken különböző állapotmódosító műveletek hajthatók végre. Ezek az ELTOLÁS, a FORGATÁS, a DEFORMÁL, a MÁSOL, a TÖRÖL, a BEHOZ, a TÁROL és az ÚJ ORIGÓ. Érdekesség a deformálás, amely nyújtást vagy zsugorítást jelent egy adott irányban.

A TRIOLA rendszer halmazművelet (például egyesítést) két test között akkor tud végrehajtani, ha azok átfedők. A kombinálásra kijelölt különálló testeket többhéjú objektumként kezeli. Közösrészképzésnél az eredményobjektumot a kijelölt két test egymással térbeli átfedésben lévő része alkotja. A különbségképzésnél a második test mintegy megcsonkítja az elsőt, vagyis az első testnek a térben a másodikkal közös része törlődik. Az ellentettképzési (komplementerképzési) művelet egyetlen testre vonatkoztatható, és eredménye az eredeti test által elfoglalt tartomány kivételével kitöltött teljes tér lesz. A műveletek redundáns jellegűek. Tulajdonképpen nincs szükség mind a négyre, mivel a többiek alkalmas kombinációjával bármelyik megvalósítható. E sajátosság a felhasználói kényelem és hatékonyság szempontjából előnyös. A kombinálásba bevont eredeti testek a halmazműveletek elvégzése után elvesznek, azaz definíciójukat a rendszer nem tárolja, viszont a kombinálás előtt tetszőleges számban másolhatók, így megőrizhetők.

Dinamikai variációk

A modellezett testek hat előre definiált, továbbá tetszőleges számú felhasználói kijelölésű pontból szemlélhetők. Bármelyik irányban síkra vetíthetők, és vetületük továbbszerkesztés (például műhelyrajz készítése) céljából átvihető a

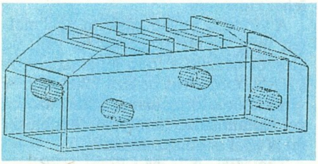


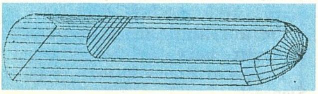
sztandard adatátviteli fájlformátumokat alkalmazni tudó 2D-s rajzolórendszerekbe. A modell képe billentyűnyomással nagyítható, illetve kicsinyíthető. A '*' billentyű 0,9-szeres, a '-' 0,5-szeres kicsinyítést, a '.' billentyű 1,1-szeres, míg a '+' kétszeres nagyítást eredményez. A nézési ablak a numerikus (kurzormozgató) billenyűkkel áthelyezhető. Egy billentyűnyomás után egy eltolási egységgel helyeződik át az ablak. Az áthelyezési lépés a NUM billentyűvel finomra vagy durvára módosítható. A görbült felületek síklapos közelítése esetén a felhasználó a szelleméleket láthatóvá tudja tenni, vagy kiolthatja azokat. A takarásban lévő élek eltávolíthatók, ezzel a testszerű hatás fokozható.

A TRIOLA lehetővé teszi a lapok színes árnyalását is. A megjelenített árnyalat beállítása a fényforrás elhelyezkedésétől függ. E tekintetben a TRIOLA szolgáltatásai még tovább fejleszthetők, bár igaz az a közmondás, hogy nem a külső, hanem a belső számít — főleg, ha háromdimenziós adatokat kell gyártási feldolgozáshoz szolgáltatni.

Partitúra — magyarul

A TRIOLA a bejelentkezése után a felhasználótól munkafájl kijelölését kéri. Ez lehet valamely korábbi fájl neve is, de az adott feladat számára megnyitandó újé is. A képernyő öt funkcionális területre van osztva. A legnagyobb hányadát természetesen a grafikus szemléltetési terület foglalja el. A grafikus terület feletti sáv a nézőpont helyzetét beállító és mutató ablak. A képernyő jobb oldalán elhelyezkedő függőleges ablak a parancsmező. A felhasználói kommunikáció (mármint a rendszerüzenetek és a felhasználói válaszok) a dialógussorban jelennek meg. A grafikus kurzor a rendszer mindenkori állapotát mutatja, azaz az utasításfogadásra kész állapotot egy kereszt, az azonosításra kész állapotot egy nyíl, a számítással való elfoglaltság állapotát egy apró homokóra jelzi. A grafikus területen kívül a kurzor egy rámutatásra emlékeztető kézként jelenik meg.





A RENDSZER menü a vízszintes (HOR FELB) és a függőleges (VERT FELB) felbontás beállítását, a behozás és a tárolás módját, test importját és exportját, valamint DOS-parancs végrehajtását teszi lehetővé. A TRIOLA használatakor automatikusan egy naplófájl képződik, amely a tervező összes lépését tárolja. E fájlok később módosítással vagy anélkül is újrafeldolgozhatók. Erre a MAKRÓ menü VÉGREHAJT opciója ad lehetőséget. A LÉPÉS paranccsal a naplófájl utasításonként léptetve hajtható végre.

Futamok egymás után

Az indítást követően a TRIOLA főmenüje jelenik meg, amely a következő opciókat tartalmazza: PRIMTEST, ÁLT-TEST, MŰVELETEK, KONTÚROK, SZÁMÍTÁS, RENDSZER, MAKRÓ és KILÉP. Az első három funkció a CSG modellezés végrehajtására, az utána következő kettő az importált geometriai elemekkel való testképezésre és a jellemző mennyiség számítására, az utolsó három kiegészítő műveletek végrehajtására szolgál. A SZÁMÍTÁS menü opcióival a testek felülete és térfogata határozható meg.

Az elemi testek méreteit azonos mértékegységben kell megadni. A biztonságos művelet-végrehajtás érdekében nem célszerű egymástól ezerszeres nagyságrendben különböző méretű testek kombinálását kérni a rendszertől. Az elemi testek méretmegadását és helykijelölését négy almenü, a PONT DEF, a HOSSZAK, a VEKT DEF és a TENG DEF teszi lehetővé. Lehetőség van abszolút és relatív értékkijelölésre is. A TRIOLA-ban a testek megforgathatók, az INS billentyű páratlan számú lenyomása a 10 fokos finom, a páros a 30 fokos durva lépésköz beállítását eredményezi.

Az instrumentumok

A TRIOLA az alapul szolgáló számítógépes konfigurációval szemben nem túlságosan igényes. Hozzá kell tenni azonban, hogy igazán jó feldolgozási teljesítményt — mint a legtöbb igazi CSG modellező — csak a munkaállomásnak megfelelő számítógépek kapacitását közelítő eszközön nyújt. Futtatásához minimálisan egy IBM PC/AT vagy azzal kompatibilis, legalább 80286-os processzorral rendelkező személyi számítógépre, matematikai társprocesszorra, a telepítéshez vagy az eredménytároláshoz 1,2 vagy 1,44 Mbájtos hajlékonylemez-kezelőre, legalább 6 Mbájt szabad merevlemezes tárolóterületre, minimálisan EGA felbontású megjelenítőre és vezérlőkártyára, valamint egérre vagy tabletre van szükség. Az adatigényes számítási műveletek miatt szükség van memóriabővítő kártya alkalmazására. Ennek logikai és fizikai kezelésére a TRIOLA szoftverbe a Lotus/Intel/MicroSoft EMS tárbővítő programját építették be.

A szoftver az MS-DOS 3.0 vagy magasabb változata alatt futtatható. C nyelven programozták, és bizonyos moduljai forrásnyelven is hozzáférhetők. Hálózati használatára jelenleg még nincs megoldás. Telepítése egyszerű, ugyanakkor az alapszintű DOS-parancsok ismeretére szükség van.

A rendszer megismerését a fejlesztők önvezérelt demóval kívánták elérni. Indításhoz csak a MOVIE parancsot kell beírni, de ne feledjük el, hogy mindaz, amit látunk, csak 'movie'. Ha magunk is hasonlók létrehozására kaptunk kedvet, a TRIOLA a RUN paranccsal bármikor zengeni kezd...

Horváth Imre

A pascalosok réme

VIGYÁZZ, UTOLÉR EGY FANTOM!

Az 1990-1991-es év fordulója friss külföldi és magyar vírusok megjelenését is hozta. Januárban és februárban folyamatosan kaptuk az információkat egyre újabb és újabb katalógusvírusok magyarországi feltűnéséről, valamint egy hazai csodabogarat is sikerült begyűjtenünk.

A tavalyi év szeptemberétől „honos" Print Screen II vírust mind több helyről jelezték a felhasználók. Ez a vírus robbanásszerűen 1990 decembere és 1991 februárja között szóródott. A vírus ténykedéséről sok felhasználó panaszkodott: hatására véletlenszerűen egy-két képernyőnyi információt kinyomtat a nyomtató. Mindazonáltal az összevissza nyomtatás nem egyértelműen a Print Screen vírus „jótette", mivel mások is megzavarják a felhasználó listázását.

Előfordult olyan eset, amelyik úgy kezdődött, hogy printerporttal rendelkező videokontrollert cseréltek le egy helyen. Az ügyes hardveres először a MIX1 vírust eresztette rá a felhasználó gépére. A vírus a printerportra küldendő karaktereket a vírus végén elhelyezett karakterkonverziós táblának megfelelően átkonvertálta mindenféle zagyvaságra. A felhasználó ezt követően a hardver felülvizsgálatát kérte. Ennek során a videokontrollert ismét kicserélték, és egyidejűleg vírusmentesítést is tartottak. A kórokozó eltávolítása és a busás számla után a rendszer jól működött tovább.

A példa nem általános érvényű, de sajnos ilyesmivel is lehet találkozni. Nem csoda, ha az ügyfelek azt kérdezik, hogy „maga hány vírust szed le, mennyit tesz fel, és mikor jön legközelebb?" Szerencse, hogy az említett, gusztustalan pénzszerzési formát nem mindenki engedi meg magának.

Magyar gengszteráru: (c) Phantom

A Phantom vírushoz a VirNet BBS-en keresztül jutottunk hozzá. A kárvallott elsősegélykérésként a NETINFO.ZIP állományt küldte fel ellenőrzés céljából.

Megvizsgálva ezt, az alábbi BATCH fájlt találtuk. (A rendszert a START.BAT programmal kell elindítani.)

```
echo off
break off
type info.txt | more
udv
```

A BACTH fájl végén az UDV.COM — Phantom vírusos — program fut le, s ezzel fertőzi meg a számítógépet.

A ZIP fájl a NETINFO cég reklámszövegét tartalmazza, amelyben felhívást intéznek a felhasználókhoz, hogyan juthatnak hozzá olcsón nemzetközi információkhoz. Nem hiszem el, hogy a NETINFO cég vírusosan adta volna ki reklámszövegét. A vírusfertőzés oka, hogy önvédelmi (önellenőrző — immune) rendszer nélkül, szöveges állományt konvertáltak át .COM kiterjesztésű programmá, majd útjára engedték. Ez a programmodul valahol megfertőződött, és így terjedhetett tovább, míg el nem jutott hozzánk. Kérdés, hogy céltudatos renomérontásról vagy a véletlen művéről van-e szó. A jelen helyzetben ezt azonban már nem lehet kideríteni.

A vírusnak egyéb tulajdonságai mellett még rejtett „üzenete" is van, de először nézzük a főbb jellemzőket. A PHANTOM vírus hossza 2203 bájt. A vírus kódolt formában az alábbi szöveget tartalmazza, amelyből részleteket fel is tálal a felhasználónak.

The PHANTOM Was HERE - Sorry... HI ROOKIE!

I'm a THESEASE! I live YOUR computer - sorry

Thanks to Brains in the Computer Siences!

A NETINFO.ZIP fájl tartalma:

Length	Method	Size	Ratio	Date	Time	CRC-32	Attr	Name
48	Stored	48	0%	11-11-90	17:49	30b5ed4e	--w	START.BAT
445	Implode	347	23%	11-11-90	17:23	8a2aaae6	--w	LEVEL.TXT
1339	Implode	893	34%	11-11-90	17:23	c6f1b5fd	--w	INFO.TXT
4227	Implode	2188	49%	15-12-90	17:55	3a67fee1	--w	UDV.COM
282	Shrunk	260	8%	11-11-90	17:13	6f344ddb	--w	MORE.COM
539	Implode	303	44%	11-11-90	18:16	c34af666	--w	RENDEL.TXT
6880		4039	42%					6

Ránk zúdul a DEN ZUK?

Betört Magyarországra is a DEN ZUK bootvírus. Az egyik felhasználó jelezte, hogy pontosan detektálni tudta. Kértük, hogy küldjön a vírusról egy másolatot, de a másolás nem sikerülhetett, mivel a vírus a kódját az eredeti bootszektorral együtt a 360 kbájtos floppylemez 40. sávjára teszi. (Közismert, hogy a DOS FORMAT programja és a DISKCOPY programok csak a 39. sávig másolnak.)

A vírus csakis a 360 kbájtos floppylemezeket szereti! Ha a vírus a memóriában aktívvá vált, akkor a „CTRL-ALT-DEL" billentyűk lenyomására grafikus üzemmódban a DEN ZUK felirat jelenik meg a képernyőn. Bizonyos számú aktivizálódás után formattálja a floppylemezt. A DEN ZUK vírusdemóprogram megismerteti az érdeklődőket a működésével.

Tajvani import: vírus a szoftveren

A vírusok terjesztéséből a számítógépek „anyaországa" sem maradhat ki. Korábban már hírt adtunk róla, hogy az Invader vírus egy közvetlenül Tajvanból beszerzett számítógép merevlemezén került be az országba. Sajnos, Tajvan potenciális, de nem tudatos vírusterjesztő lehet. Most februárban egy modem szoftverével hurcolták be a Music Bug v.1.02 vírust hozzánk. Akkor csak a SCAN 71 program tudta ezt a kórokozót felismerni, azonosítani.

A Music Bug egy bootszektorvírus: a merevlemezt megfertőzve a FAT táblában 4096 bájt láncolási hibát jegyez be, és ezeken a szektorokon tárolja magát. Képes egymás után is egyre több láncolási hibát létrehozni a lemezen, csökkentve a szabad lemezterületet. A vírus aktivizálódása során zenél. (Úgy látszik, a zenei tehetségű vírusok lettek népszerűek!)

Copyright (c) PHANTOM -- This virus was designed in
HUNGARIAN VIRUS DEVELOPING LABORATORY. (H.V.D.L.)
v1

A vírust egyelőre az ország közepén észlelték: Budapesten, Vácott. Június 23-ig csak a .COM programok érdeklik. Ezt követően az .EXE programokat a hét napjától függően, különböző formában fertőzi meg. Minden fertőzési napon az alábbi plusz bájtokat teszi a Copyright sztringjére:

Hétfő: 8
Kedd: 7
Szerda: 9
Csütörtök: 14
Péntek: 8
Szombat: 10
Vasárnap: The PHANTOM Was HERE - Sorry...

De ez még nem minden! Az .ARC és .DBF állományok elejére a Copyright sztringjét írja be. A .DBF állományok elején az adatbázis fejléce, vagyis a struktúra szerkezete van; ezzel az adatbázis használhatatlan lesz. Az ARC állományok olyanok, hogy az egyes tagokból lehet kiszámítani a következő elemet, ezért minden .ARC állomány meg fog sérülni.

A vírus kimondottan a pusztító fajtájúak közé tartozik. Az egyszerűség kedvéért a videokontroller kikapcsolását is elvégzi. Az egyik felhasználó a kép visszakapcsolására már külön kis programot is írt, mire rájött, hogy a háttérben ott a vírus. Mindenesetre előtte egy pár napig szenvedett...

A vírus a következő interruptokat cseréli le: 20, 21, 24, B2. A bootszektorral is manipulál valamit. Visszafejtése még nem fejeződött be teljesen, de már az biztos, hogy INT 25-tel, INT 26-tal kezeli.

A dolgok pikantériája

A VI. 23. Farmosi István kollégám születésnapja. Farmosi előszeretettel használja szoftvereiben saját, illetve családtagjai személyi számának részleteit. Így került A PRGDOKI v.2.11 belsejébe is egyedi azonosítóként a 06.23-as. A vírus visszafejtése közepette Farmosi meglepetten közölte velem, hogy az ő születési dátumával találkozva a vírus másképpen viselkedik.

Kétségtelen, hogy a vírust nem Farmosi írta, de szinte az is biztosnak mondható, hogy az ő hírneve befeketítésére szánták. Ezt Veszelin Boncsev hasonló esetével lehet párhuzamba hozni, akinek a nevét a v2000 vírusban helyezte el ötletes szerzője. Veszelin Boncsev szintén ártatlan volt. Ezek a tények megerősítik, hogy a személyre vonatkozó adatokat (név, becenév, személyi szám, telefonszám) és a családtagok vagy a közvetlen környezet által használt szójárásokat miért nem szabad azonosítóként, illetve jelszóként megadni.

A vírusírás (átírás) lélektana

Az elmúlt időszak számítástechnikai érdekessége, hogy jópofa, „menő" témának számít a vírusírás. Ez egyeseknek szakmai kihívás (ki ír jobb vírust, kié terjed el gyorsabban), másoknál a fogadás a lényeg, és vannak kitolás jellegű ambíciók is. Tapasztalataink szerint azoknak a vírusoknak a klónjai (átiratai) jelennek meg Magyarországon, amelyekről a szakma sokat beszél és a sajtóban is nagy a publicitásuk. A vírusírók a vírusölők megtévesztésére, mások munkájának tönkretételére használják fel szakmai kvalitásaikat. Ha képességeiket az adatbiztonság érdekében kamatoztatnák, az sem volna kisebb dicsőség!

Az Anti-Pascal vírus

A vírus neve: Anti-Pascal.
Egyéb elnevezése: Anti-Pascal 605 vírus, AP-605, C-605, V605.
A vírus státusza: kereső.
Az első észlelés időpontja: 1990. június.
Tünetei: .COM hossza nő, .BAK és .PAS fájlok tönkremennek.
Eredete: Bulgária.
Izolálási helye: Szófia.
Hossza: 605 bájt.
Típuskód: PNCK (parasitic, non-resident, .COM infector) parazita, nem rezidens, .COM-fertőző.

Az Anti-Pascal vírust (V605, illetve C-605) 1990 júniusában azonosította Veszelin Boncsev Bulgáriában, Szófiában. Eredetileg úgy gondolták, hogy az Anti-Pascal vírus a Szovjetunióból vagy Lengyelországból származik.

Vírus — batch-fájlokban!

A vírusok eddig szokatlan típusa a batch-fájlvírus. Amíg nem találkoztam ilyen példánnyal, számomra is elképzelhetetlen volt batch-fájlokban vírust keresni. Most volt min elcsodálkoznom! A batch-fájl közepében egy gépi kódú rutint helyeztek el, amit az alábbi példában — lényegesen megcsonkítva — bemutatok. A vírusgazda fájl elindítása után a kórokozó megfertőz minden batch-fájlt.

```
@echo off
if exist bug.com goto infection
echo 1+il█G1L=!>bug.tmp
copy bug.tmp /a bug.com >nul
for %%a in (*.bat) do call %0 %%a
del bug.*
goto start
:infection
bug <%1 >nul
if errorlevel 255 goto end
bug <%1 >bug.tmp
bug <%0 >%1
copy %1+bug.tmp %1 >nul
echo :end>>%1
goto end
:start
echo on
:@echo        Don't panic !
@echo Eddie lives ... somewhere
              in time !
@echo   Welcome to the Dungeon
@echo   I wish you luck !
:end
```

DECNET vírus?

Az elmúlt időkben feldúsult vírusáradat során egy ismeretlen formátumú .COM fájl került a kezünkbe. Ezt a programot DECNET vírus néven juttatták el hozzánk. Ami bizonyos: ez nem egy futtatható COM program. Várjuk azok jelenkezését (122-3025, Szegedi Imre), akik a következő programrészlet alapján ráismernek a programozási verzióra. Előzetes információink szerint elképzelhető, hogy a DOS alatt ismert „Father Christmas" vírus DEC-es átirata lehet.

```
$ MAILLINE9 = " I OFTEN FIND A LITTLE PRESENT OFFERED BY
THE CHILDREN,"
$ MAILLINE10 = " OR EVEN A LITTLE BRANDY FROM THE FATHER.
(YEAH!)"
$ MAILLINE11 = " ANYHOW THE CHIMNEYS ARE GETTING TIGH-
TER AND TIGHTER"
$ MAILLINE12 = " EVERY YEAR. I THINK I'LL HAVE TO PUT MY
DIET ON AGAIN."
$ MAILLINE13 = " AND AFTER CHRISTMAS I'VE GOT MY BIG
HOLIDAYS :-)."
$ MAILLINE14 = ""
$ MAILLINE15 = " NOW STOP COMPUTING AND HAVE A GOOD
TIME AT HOME !!!!"
$ MAILLINE16 = ""
$ MAILLINE17 = "    MERRY CHRISTMAS"
$ MAILLINE18 = "     AND A HAPPY NEW YEAR"
$ MAILLINE19 = ""
$ MAILLINE20 = "              YOUR FATHER CHRISTMAS"
$ NUM_MAILLINES = 21
$ DEFINE SYSUAF SYS$LOGIN:SYSUAF
```

Megállapították azonban, hogy Bulgáriában írtak egy keresővírust több mint egy évvel korábban. A szerző egészen 1990 júliusáig nem is tudott a „szökésről".

Az Anti-Pascal vírus a .COM fájlokat fertőzi, beleértve a COMMAND .COM-ot is. Mivel nem memóriarezidens, a fertőzés során a 24 interruptot lefoglalja.

Az Anti-Pascal vírussal fertőzött programok végrehajtásakor a vírus két további .COM fájlt is megtámad — vagy az aktuális meghajtón, vagy a D: meghajtón. Ezek olyan állományok, amelyek hossza 605 és 64 930 bájt közötti. Ha a vírus talál a kiválasztási kritériumainak megfelelő, fertőzetlen .COM fájlt, akkor a program első 605 bájtját felülírja a víruskóddal, a program eredeti 605 bájtját pedig a már megfertőzött fájl végéhez fűzi. A fertőzött állományok hossza tehát 605 bájttal megnő, a „PQVWS" karaktersorozattal kezdődnek, és a 0x17 eltolásnál a „combakpas???exe" sztringet tartalmazzák. A beteg állományok dátum- és időbejegyzései is módosulnak a könyvtárban: a fertőzés dátumára és időpontjára.

Ha az Anti-Pascal vírus nem talál két olyan .COM állományt, amelyet meg tudna fertőzni, tovább keres az aktuális meghajtón .BAK, illetve .PAS állományokat. Ha vannak ilyenek, akkor ezeket felülírja a víruskóddal. Ha a felülírt állomány .PAS kiterjesztésű volt, akkor a forrásprogram egy része elveszett. Miután a vírus felülírta a .BAK, illetve a .PAS állományokat, megpróbálja átnevezni ezeket .COM-má. Ha van már ilyen .COM állomány, akkor .EXE-vé nevezi át a fertőzött fájlokat. Ez az átnevezés azonban nem működik — egy vírushiba miatt!!

Az Anti-Pascal vírus ismert variánsai: AP-400, AP-480, AP-529, AP-605

Az AP-529 a 605 bájtos Anti-Pascal vírushoz hasonló, az a fő különbség, hogy az AP-529 csak 2048 bájtnál hosszabb .COM fájlokat fertőz. A fertőzött állományok hossza 529 bájttal nő. A .BAK, illetve .PAS fájlokat pedig nem felülírja, hanem kitöröl egy .BAK vagy .PAS fájlt, ha nem talált az aktuális meghajtón egészséges, 2048 bájtnál hosszabb .COM fájlt. A C: meghajtó főkönyvtára akkor is megfertőződhet ettől a vírustól, ha a végrehajtás az A: vagy a B: meghajtóról történik. Ez a variáns „kereső" (research) vírusnak minősítendő, nem célszerű tehát, hogy nyilvános legyen.

Szegedi Imre

COM-korlát

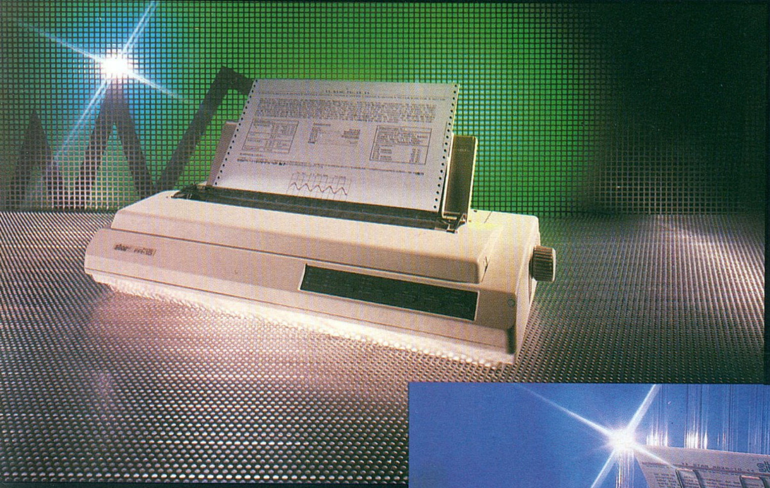
Nyugati szakmabeliek véleménye szerint nálunk az iskolákban a diákok tananyagként tanulják és szakmai gyakorlatként kapják a vírusírást. Azért a környező országok számítástechnikai kultúrája is kibontakozóban van — egyre több lengyel és szovjet vírus jut el hozzánk... Az egyik ilyen érdekesség a „LENINFORMATIKA" hardverjavító cég vírusba ágyazott reklámja. Ez a csaknem 24 kbájtos vírus grafikusan propagálja az említett céget. A vírus aprócska hibája, hogy 130 kbájtos a COM program. Úgy látszik, a szovjet vírusírók egyelőre figyelmen kívül hagyják a korlátokat. (Mármint, hogy egy .COM program legfeljebb 64 kilobájtnyi lehet.)

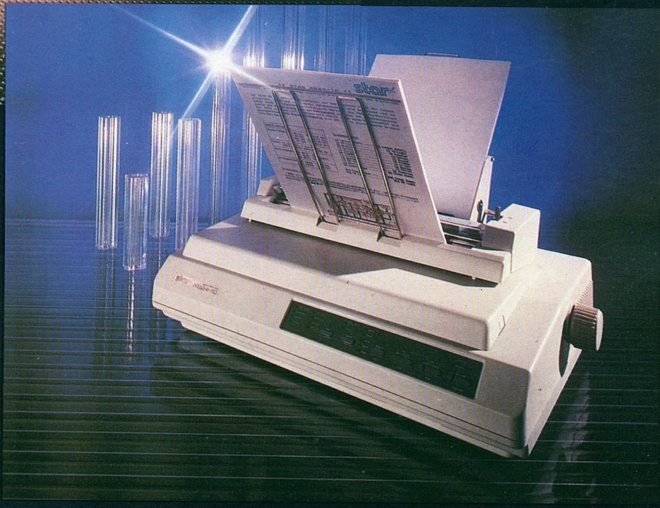
Új víruskereső: a PCSCAN

A víruskereső/vírusölő programok kategóriájában újabb programcsalád első tagja látott napvilágot. A PCSCAN program kereskedelmi verziója 203 vírus felismerésére alkalmas víruskereső program (a SCAN 72 csak 156-ig jut el). Ha figyelembe vesszük az egyes mutánsok számát a „megszokott" vírusmutáns/darab számolási móddszerrel, akkor a programmal megközelítőleg 300 vírus felismerésére számíthatunk. Ez a program — ellentétben más nyugati víruskeresőkkel — már megtalálja a szovjet produktumokat is. A program fejlesztésénél a sebesség, nem utolsósorban az időrabló képernyőkezelés javult. A program installálásához — Windows 3.0 alá — a lemez tartalmazza a szükséges fájlt. A programhoz floppy méretű kezelési utasítás is készült. A víruskereső program sok opcióval rendelkezik, köztük a naplókészítéssel is. A program 1000 Ft-os áron megvásárolható az Ázsió-Microtrade Kft.-nél, a Floppyland-ben és a Cédrus Rt. megbízott viszonteladóinál.

star

the ComputerPrinter





INFORMÁCIÓKÉRÉS: 40 ▼

Exclusive Distributor:

HRP consultants S.A.R.L.

Keleteurópai Kereskedelmi Képviselet és bemutatóterem
1051 Budapest V., Nádor u. 32. Telefon: 132-1811, 132-7534 Fax: 131-8177

SZERVIZ:
1055 Budapest
V., Balassi B. u. 25.
Tel.: 131-0044, 131-3980

SIEMENS

Cégünk 100 éve tevékenykedik Magyarországon. Legújabb sikerünk: részt veszünk a telefonhálózat fejlesztésében.



Magyarországnak új telefonhálózat –
Önnek új telefon alközpont.

Telefon alközpontjaink a csúcstechnológiát képviselik. Az optimális megoldást nyújtjuk kis és nagy vállalkozások részére.

Forduljon hozzánk bizalommal!
Látogasson meg bennünket:

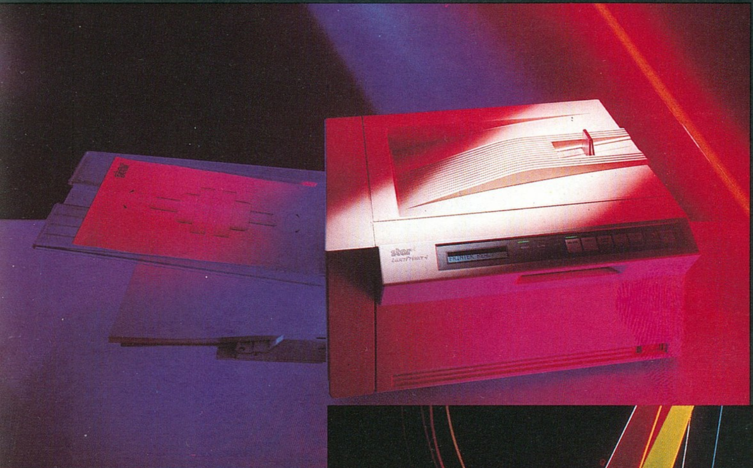
ifabo Budapest: 1991. május 7–10.

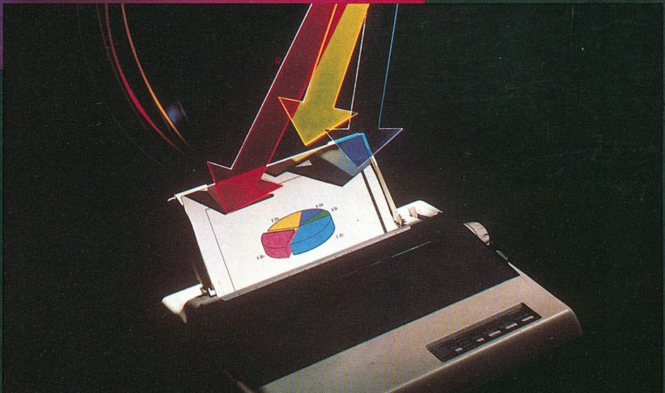
Siemens KFT Budapest
H-1115 Budapest, Bártfai u. 54.
Telefon: 186-8044
Fax. 185-3261
Tx: 224133

A mi tapasztalatunk – az Ön haszna!

star

the ComputerPrinter





Exclusive Distributor:

HRP consultants S.A.R.L.

Keleteurópai Kereskedelmi Képviselet és bemutatóterem
1051 Budapest V., Nádor u. 32. Telefon: 132-1811, 132-7534 Fax: 131-8177

SZERVIZ:
1055 Budapest
V., Balassi B. u. 25.
Tel.: 131-0044, 131-3980

„Gyorsjáratú" programkötegek

Sok felhasználó bosszúságára az MS-DOS nagyobb programkötegeinek végrehajtása elég lassú. Ha az állományokat a BAT2EXEC.COM Batch-Compiler segítségével fordítjuk le, kiküszöbölhetjük ezt a hátrányt.

Programkötegek (batch-fájlok) nélkül ma már elképzelhetetlen a személyi számítógépeken végzett mindennapi munka. Bizonyos idő elteltével minden felhasználónak van belőlük saját külön kis gyűjteménye is. Vannak, akik többezer bájtos „műalkotásokat" hoznak létre, eljutva egészen a batch-nyelv lehetőségeinek határáig. A funkciók kibővítésére időközben sok segédprogram is készült, mint például a BATCHMAN.COM, ami egyes felhasználókat még terjedelmesebb programkötegek írására ösztönzött.

Sajnálatos módon a programkötegek motorikusan lassúak. Felgyorsításuk legegyszerűbb módja az átfordítás. Ebben játszik nagy szerepet a BAT2EXEC.COM program, amely elolvassa egy batch-fájl ASCII szövegét és abból .COM formátumú programot készít az eredeti programköteg által elvégzendő feladat megoldására. Az új program ugyanúgy végrehajtja a belső DOS parancsokat, elindítja a felsorolt programokat, kiértékeli a környezeti változókat és a parancssorok érveit. Az egyetlen lényeges különbség: a sebesség.

A program kezelése nagyon egyszerű: ki kell adni a BAT2EXEC FÁJLNÉV.BAT parancsot. Ha a programköteg valamelyik sorát a fordítóprogram nem érti, akkor hibaüzenetet küld és megjelöli annak a sornak a számát, amelyikben a hibát felfedezte.

A program működésének megértéséhez szem előtt kell tartani a fordító (compiler) és az értelmező (interpreter) programok közötti különbséget. Mindkettőhöz szükséges egy kiinduló forrásprogram valamilyen meghatározott nyelven (pl. C, Basic vagy az MS-DOS batch-nyelve). Ezzel azonban ki is merült a hasonlóság. Az értelmezett (interpretált) programban egy másik program, az interpreter, soronként elolvassa és rögtön végrehajtja az ott található parancsokat. Ha tehát egy sor ismétlődik, többször kell megismételni minden műveletet. Gyakran pedig egy parancssor elolvasása és dekódolása ugyanannyi időt vesz igénybe, mint annak végrehajtása, mert sok mindent kell megvizsgálni, míg egyértelműen kiderül, hogy a parancs kiadható.

Az értelmezett (interpretált) batch-fájl legnagyobb előnye kétségtelenül az, hogy azt a felhasználó rögtön megírása után el is indíthatja, bármelyik ASCII szövegszerkesztővel gyorsan elkészítheti, azonnal kipróbálhatja, a hibát könnyen korrigálhatja.

Az interpretálttal ellentétben a lefordított (kompilált) batch-program önállóan futtatható. Az ASCII forrásszövegek parancsainak felismeréséhez szükséges munkát a fordításkor a fordítóprogram elvégzi. Ha ugyanaz a parancs egy ciklusban többször előfordul, nincs szükség újabb parancsfelismerésre. A végeredmény olyan program, amelyben csak a végrehajtáshoz szükséges gépi parancsok vannak.

Természetesen a Batch-Compiler nem „csodaszer": a program megírása így tovább tart; a forrásszöveget módosítva a fordítást mindig meg kell ismételni; a kész programban a hibakeresés megfelelő segédprogramok nélkül nagyon nehéz, mert gépi kódból kellene olvasni.

A programot nem szabad minden esetben alkalmazni. Például az AUTOEXEC.BAT tilalmi listán van, mert a DOS-parancsértelmező COMMAND.COM egyébként nem találja meg. A memóriarezidens TSR programok pedig ebben a lefordított állapotban futtatás után nem tudnának beülni a memóriába, mert az ehhez szükséges felületet a BAT2EXEC.COM saját részére lefoglalja. Nem tilos, de nem is célszerű átalakítani rövid kis batch-programjainkat, hogy azokba bármikor könnyen bele tudjuk tekinteni és bele tudjunk javítani. Annál hasznosabb viszont a BAT2EXEC.COM olyankor, ha monstre programkötegeinket akarjuk „gyorsjáratúvá" tenni.



(PC Professionell, 1991/április)

Vissza a kézíráshoz

Bill Gates, a Microsoft főnöke arról nevezetes, hogy új ötleteivel mindig egy lépéssel mások előtt jár. Az sajnos már valamivel tovább tart, amíg elképzeléseit sikerül a gyakorlatban kivitelezni. Most is ez játszódik le, mert bár egy évvel ezelőtt beígérte a kézírást olvasó programot, annak konkrét körvonalai még csak most kezdenek kibontakozni egy Windows-bővítés formájában.

Windows-H elnevezéssel néhány fejlesztő hónapokkal korábban megkapta a Pen-Windows programot, amelynek a következő Windows-verzióval együtt kellene piacra kerülnie, s amelynek képesnek kell lennie a kézírás felismerésére és kiértékelésére. Túl sokat egyelőre azonban még nem szabad várni: a felhasználó egy speciális olvasóval kézírást tud majd bevinni számítógépbe, s az írásfelismerés egyszerű jelekre és parancsokra korlátozódik. Szó sincs tehát még igazi, univerzális kézírásolvasásról.

Egy kicsit messzebbre jutott ezen a téren a Go Inc. (Foster

City, USA), saját PenPoint („Tollhegy") rendszerével, amely jóval nagyobb felismerőképességgel rendelkezik. Egyetlen hátránya, hogy az MS-DOS-hoz nem illeszkedik, az alkalmazáshoz az alapoktól induló új fejlesztésre van szükség.

A Pen-Windows-hoz várhatóan csak 1991 nyarán készülnek el az első fejlesztő-készletek. A Microsoft nyilvánvalóan azért sietett az új szoftver bejelentésével, hogy előtérbe állítsa azt, szemben a Go Inc. nagyobb teljesítményű, konkurens termékével.

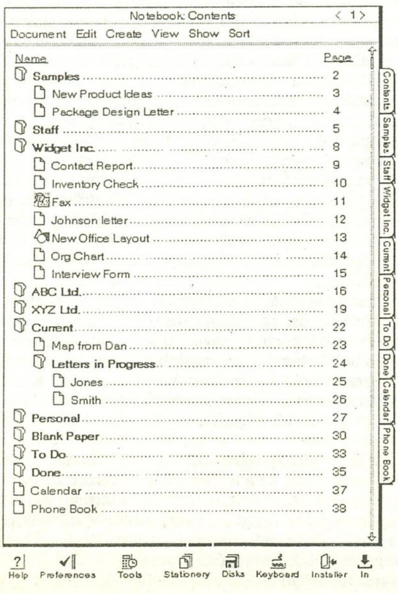
A verseny akkor válik majd valóban izgalmassá, amikor a hardvergyártóknak dönteniük kell, hogy melyik változatot alkalmazzák és teszik általánossá. Az IBM állítólag már a PenPoint javára (tehát a Microsoft hátrányára) döntött, míg az NCR, a Wang Labs és a Kyocera najd a Microsoft termékét részesíti előnyben.

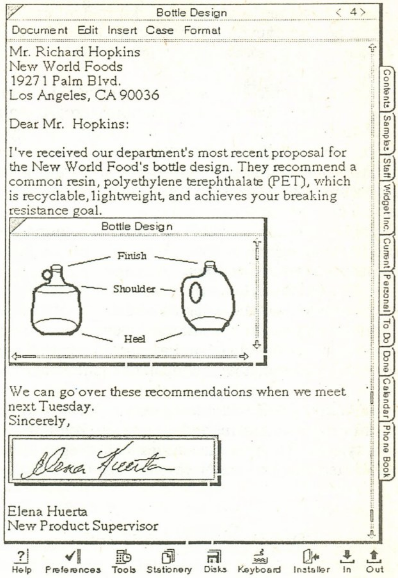
(PC Professionell, 1991/április)

Tollhegyre tűzve

A PenPoint egy 32 bites, tárgyorientált, többfeladatú (multitasking) operációs rendszer, amelyet speciális tollal kezelhető, mobil számítógépek különleges igényeinek kielégítésére fejlesztett ki a Go Corporation.

Az új szoftver egyik legfontosabb eleme a jegyzettömbként felfogható alkalmazási felület (NUI = notebook user interface). A tömb lapjai ugyanabban a formában őrzik meg az információkat, ahogy azokat felvittük, illetve ahogy utoljára módosítottuk. Mivel többszáz vagy többezer jegyzetlap szimultán futtatása nagyon körülményes lenne, a program csendben elvégzi a könyvelési munkákat: lapozáskor mindig





elmenti, illetve betölti az adatokat. (Hogy illúziónk teljes legyen, a valódi lapozáshoz hasonló grafikai hatás kíséretében!)

A jegyzettömb első oldalán van a tartalomjegyzék, amely nagyon hasonlít a papírból készült elődökhöz. (Lásd a mellékelt illusztrációt.) A tollal a megfelelő lapszámot megérintve a program azonnal odaugrik. A lapokat tetszés szerint lehet válogatni, csoportosítani — ahogy azok információit éppen feldolgozni akarjuk. A PenPoint képernyő alján van a „könyvespolc", ikonokkal jelölve a különböző funkciókat.

Az információ rögzítése elsősorban a tollal történik, de billentyűzet alkalmazására is van alternatív lehetőség. Képernyőkurzorra nincs szükség, mert a toll hegye meghatározza a helyet a képernyőn. A tollat lehet használni jelölésre (megérintve a képernyőt), adatbevitelre (kézírással) és utasításra (tollmozdulatokkal). Az utóbbiak a hagyományos szerkesztési jelrendszerre támaszkodnak.

Miközben a képernyőre kézzel írunk, a PenPoint program a háttérben elvégzi a jelfelismerő műveletet, és a továbblapozott oldal firkálmánya ASCII szöveg formájában azonnal készen áll a további feldolgozásra. Ez úgy működik, hogy egy alrendszer minden apró tollmozdulatról alacsony szintű üzenetsorokat generál, majd azokat magasabb szintű egységekbe, „firkákba" (scribbles) csoportosítja. A firkákat pedig egy kézírásfordító (HWX) alrendszer szövegkarakterekké vagy tollmozdulat-parancsokká alakítja.

A jelenlegi kézírásfordító az írás műveletével egyidejűleg a háttérben működik, 286-os processzor és 16 MHz órajel esetén másodpercenként 3 betű sebességgel, a memóriából pedig kb. 100 kilobájtot foglal le. Másik 100 kilobájtot vesz igénybe egy általános elválasztási szótár. A HWX számára nem jelent akadályt, ha a betűk összeérnek, egymásba fonódnak vagy töredékesek, sőt azt is elfogadja, hogy ugyanaz a személy ugyanazokat a betűket különböző időpontokban másként írja. A felhasználók rövid gyakorlás után eljutnak

oda, hogy a program 5 szóból 4-et korrekten értelmez, az ötödikben pedig könnyen javítható, apró hibát vét. (Ez szavakra vetítve 80-90 százalékos, karakterekre vetítve 90-97 százalékos pontosság.)

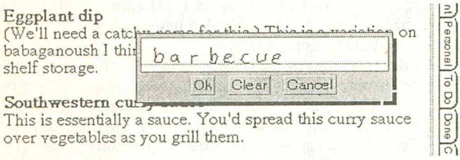
A jelfelismerés művelete során az információ több forrás kombinált felhasználásával és dinamikus program-algoritmussal elrendezve jön létre. Az ismeretforrások egy része rögzített, mint például a 100 000 szavas szótár vagy az írásjelhasználat szabályzata, más részük az alkalmazás során kialakuló speciális összefüggésekre utal, például speciális listákból áll.

Az egyik legfontosabb belső „tudásforrás" természetesen a karakterek alakjának felismerésére szolgál. A felismerő mechanizmus a jelként beérkező „firkákat" elemzi, összehasonlítja őket az egyes karakterekre vonatkozóan mintaként tárolt sokszáz alakzattal. A legtöbb kézírás elolvasásához a mintakészlet elegendő is. Amikor mégsem, akkor rövid írráspróbával saját macskakaparásunkat is betáplálhatjuk a gépbe, és így testre szabottan növelhetjük annak jelfelismerő képességét.

A PenPointhoz a Go által kifejlesztett hardver ugyanúgy néz ki, mint egy standard méretű, vastag jegyzettömb — amíg ki nem nyitjuk. Kinyitva viszont leginkább egy laptop gép folyadékkristályos képernyőjére emlékeztet. A képernyő kb.

CORE PENPOINT GESTURES

Tap	•	Select/Invoke
Press-hold	◉	Initiate drag (move, wipe-through)
Tap-hold	• ◉	Initiate drag (copy)
Flick (four directions)	|	Scroll/Browse
Cross out	X	Delete
Scratch out	═	Delete
Circle	O	Edit
Check	✓	Options
Caret	Λ	Insert
Brackets	[ ]	Select object, adjust selection
Pigtail (vertical)	℘	Delete character
Down-right	L	Insert space



14 cm széles és 20 cm magas. A leginkább mérnöki ceruzára hasonlító, vezeték nélküli toll elektromágneses mezőt generál, amely a képrenyő felületén áthatol, így a rendszer érzékeli, hogy hol tartózkodik és hogy érinti-e a képrenyőt. A PenPoint rendszer használata valószínűleg sok memóriát követel majd, lehet, hogy 4 Mbájtot is. Ennek ellenére a feldolgozott adatokat elég tömören kezeli, mert 1 Mbájtnyi RAM akár 100 oldalnyi szöveges és grafikus adatot is tárol.

A PenPoint tulajdonképpen egy paradigma-váltás előhírnöke. Teljesen kiküszöböli a számítógépes adatbevitel eddigi általános eszközét, a billentyűzetet. A rendszert használva az lehet az érzésünk, mintha valóban papírra, jegyzettömbbe írnánk. A felhasználó előtt megnyílik az a perspektíva, hogy megőrizze korábbi jegyzetelési szokásait, sőt az általa leírtakat vázlataival, rajzaival kiegészítse, eredetiben bemutassa. Ugyanakkor arra is lehetősége lesz, hogy az egészet személyi számítógépen feldolgozza (pl. DTP) vagy azzal adatbázisának korábbi állományait korrigálja.

Ha a szoftverfejlesztés a mostani ütemben halad, és a hardvergyártók elég gyorsan reagálnak erre a kihívásra, akkor a lyukkártya, a lyukszalag, a billentyűzet, a képernyő, az egér után az ember és a számítógép közötti kommunikáció hamarosan olyan új eszközrendszerrel bővül, amely igazán emberközelivé teszi a számítástechnikát.

(Byte, 1991/február)

Stílusjavító programok

A nyelvtani és stíláris javítóprogramok közül kettő vált különösen népszerűvé: a Grammatik IV és a RightWriter. Csak az a gond velük, hogy miközben jól kezelik az általános angol nyelvet, nem sokat segítenek az egyéni és rétegstílusok kialakításában. A legtöbb cég például igyekszik saját stílust kialakítani vagy alkalmazkodik vevőinek stílusához, különösen, ha azok kormányzati szervek. Néha egy helytelen szóhasználatnak is költséges következményei lehetnek, ezért érdemes a kimenő dokumentumok nyelvtani és stiláris hibáit minimálisra csökkenteni.

Két szoftver, a Grammatik IV Government Edition és a Corporate Voice próbálja a nyelvnek speciális környezetben való használatát megoldani, de módszertanilag alapvetően különböző módon közelítik meg a kérdést. A Corporate Voice a javítandó írást meghatározott dokumentumokkal veti össze és megmutatja az eltéréseket. A Grammatik IV Government Edition a kéziratot a stiláris és nyelvhasználati vezérfonalak szempontjából elemzi és ad javaslatokat a változtatásokra. A kreativitást egyik program sem fékezi, hanem arra ösztökéli a szerzőket, hogy meghatározott normáknak megfelelő módon változtassanak a stíluson.

A Corporate Voice a dokumentumot egészében vizsgálja és jelzi, hogy az hogyan viszonyul a „mintákhoz". A mérce kiválasztásában szabadon dönthetünk, tehát nem kell attól félni, hogy a program ránk erőltet tőlünk idegen megoldásokat. Olyan stílusmintákra lehet támaszkodni, mint általános, műszaki, tudósító vagy magazin stílus, de akár bizonyos írók regénystílusa is választható. Ha saját stiláris követelményeket akarunk meghatározni, akkor ki kell jelölnünk azokat a forrásműveket, amelyek alapján a számítógép a normarendszert felállítja.

A stílusilleszkedést grafikusan is ábrázolni lehet. A standard szövegstílust egy könnycsepp formájú grafika jeleníti meg. Ha az ellenőrzött szövegről kirajzolódó forma ennek határain belül marad és a mondathosszúság vagy a szóhasználat paraméterei a mintához közeliek, akkor szövegünk stílusa konzisztens. Ez persze még nem jelenti azt, hogy amit írtunk, az tényszerűen igaz is, vagy hogy az egész egyáltalán jól van megírva.

A stílusellenőrző programok révén nyilvánvalóan senkiből nem lesz nagy író, de a tapasztalatlan szerzők íráskészségét eredményesen csiszolhatja, önálló stílusuk kialakulását elősegítheti. S ami a legfontosabb: a rutinos íróknak is megkönnyíti a szerkesztési munkát, több időt fordíthatunk a szöveg tartalmi összefüggéseire.

(Byte, 1991/február)

Újdonságlesen

Európa legnagyobb számítástechnikai és elektronikai szakvására a hannoveri Cebit. A rengeteg kiállított terméket alaposan végigböngészni talán a vásár egész időtartama alatt is képtelenség volna. (A legnagyobb pavilonnak kiállításra használt két emelete nagyobb alapterületű, mint a Budapesti Nemzetközi Vásáré!) A tájékozódásban viszont segített a kiváló számítógépes információs rendszer — meg az újságírói szubjektivitás és szimat.

A külső szemlélő számára is nyilvánvaló: a számítástechnika fejlődése nem egészen a jóslatok szerint alakult. Továbbra is stabil lábakon áll a hagyományos MS-DOS operációs rendszer, ugyanakkor rohamosan terjed a UNIX, s mintha egy kicsit ismét meglódult volna az OS/2. Bár ez utóbbi érzés onnan is származhat, hogy az IBM minden áron szeretett volna uralkodni a kiállításon. Pénzt, paripát nem kímélve mindegyik tematikus kiállításán ott nyílott a „kék rózsa". Olyannyira, hogy az IBM egy kétszáz oldalas külön katalógust is kiadott, hogy mit, hol lehet megtalálni termékeiből a hatalmas számítástechnikai tengerben.

A mikroprocesszortechnika az Intelnél és az Intelen kívül sem állt meg a fejlődésben. A Cebit ideje alatt jelentette be az amerikai Advanced Micro Devices cég, hogy elkészítette az Intel 80386-os processzorral szoftveresen és elektronikusan is teljesen kompatibilis, ugyanakkor egészen más belső áramköri elvekre épülő klónt. Az új chip 44 százalékkal kisebb az Intel hasonló áramköri lapkájánál. Jele AMD 386DX. Megjelentették ennek energiatakarékos verzióját, az AMD 386DXL processzort is, amelyet főleg laptop gépekbe és jegyzettömb mértetű AT-kba javasolnak beépíteni. Mindegyikből 25 és 35 megahertzes órajelű változat készült el eddig, de készül a 40 megahertzes változat is. Az energiatakarékos processzor áramfelvétele készenléti üzemben 1 mA, míg az Intel chipnél ez 386-tól 550 mA-ig terjed.

Az Intel is tartogat fejlesztési meglepetéseket a közeljövőnek. Csak az a kérdés, hogy nem kapkodják-e el szokás szerint a dolgokat. Ugyanis eddig minden processzorsorozatuk első tagjai hibásak voltak. A most megígért új processzor az Intel 386-tal lesz kompatibilis, de 250 megahertzes órajellel, 2 megabájtos gyorsítótárral, a grafika számára két vektorgrafikus beépített segédprocesszorral, 2000 MIPS számítási teljesítménnyel. Most már nemcsak a matematikai koprocesszort tartalmazza a chip, mint a 486-osoknál, hanem a Weitek rasztergrafikus processzort is!

A kiállítás másik számítástechnikai érdekessége volt, hogy korábban lebecsült gépek és gyártók professzionális termékekkel az MS-DOS gépek teljes értékű helyettesítőjévé, komoly kiegészítőjévé váltak. A Commodore például UNIX-alapú munkaállomásaival jelent meg. Bizonyos jelek arra utalnak, hogy az Amiga hasonló karrier előtt áll, mint a Macintosh. Főleg animációs szoftvereket, reklámfilmstúdiókat és CAD/CAM-munkahelyeket fognak kialakítani a segítségével. Ezt segíti, hogy a professzionális televíziózásban a kép és a hang illesztése tökéletesen megoldott. Ugyanakkor a szoftverházak is egyre inkább gondolnak arra, hogy nemcsak MS-DOS-felhasználók vannak a személyi számítógépek használói között. Ennek a törekvésnek a jegyében készítették el a PC-s világban (vagy máshonnan ismert) szoftverek adatkompatibilis, másgépes verzióit. Ebben a fejlődésben a motor szerepét a Microsoft látta el, amikor elkészítette a közkedvelt Word szövegszerkesztő Macintosh-verzióját, amit nagyon gyorsan követett az UNIX alá írt változat. Most pedig már egyre több DTP-szoftver él ilyen kettős életet. A Xerox Ventura Publisher is elkészült Macintoshra és OS/2-re, most pedig készítik az UNIX-verziót.

Az új Ventura Publisher-család a keresztségben a Gold sorozatnevet kapta. Másolásvédelem nélkül kerül forgalomba. A Windows, a GEM/DOS és a Macintosh-verzió a gyártó tájékoztatása szerint az adatok szintjén teljesen kompatibilis. Az ottani programozó mérnök megerősítette azt a tapasztalatot is, hogy a 3.0-s és a 2.0-s Ventura-verziók nemcsak az adataik, hanem belső meghajtóik, betűkészleteik szintjén is kompatibilisek, de ez a megállapítás csak a DOS/GEM verzióra igaz! Nem kell tehát beszerezni hozzájuk új monitormeghajtó programot, printermeghajtót, betűket vagy elválasztási algoritmust. Tapasztalataink szerint ez mindaddig érvényes amíg nem kell a magyar ékezést alkalmazni. Ha ugyanis nem német vagy USA karakterkészletes változatot alkalmazunk, akkor csak az adatlapok kompatibilitása áll fenn, a szöveget konvertálni kell saját segédprogramokkal.

Mindenesetre a Gold sorozat tagjai teljesértékű programok. Egyetlen installálókészlet tartalmazza a normál (vagyis az EMS használatát nem igénylő), a professzionális (azaz az EMS-t igénylő, táblázatszedésre is alkalmas), valamint a hálózatkezelő interfészt. Ez utóbbi a gyártó tájékoztatása szerint csak a Novell 3.1 alatt működik jól Novell-módban. Hogy mit ajánl a régebbi verziót használóknak? Cseréljék ki az ANET3-at vagy ANET4-et az újabbra, akkor nem kell lecserélni a 2.XX-es Novellt, de egyes funkciókkal zavarok lehetnek, bár az nem bizonyos... Annyi bizonyos, hogy a cég nem szívesen ad barkácsolási tanácsokat.

A Word Perfect esetében már nem kellett olyan gondokkal számolni. mint a Venturánál. Ugyanis ez a világ egyetlen olyan DTP-szövegszerkesztő programja, amely az összes latinbetűs nyelvre, így a magyarra is fel van készítve. A probléma mindössze annyi, hogy az ékezetes betűket csak postscript üzemmódban képes produkálni nyomtatón... S ha már a DTP-nél tartunk: ebben is forradalmi változások történtek az utóbbi évben. Az alapszoftverek gép-internacionálissá váltak, azaz a programverziók futtathatók és adatkompatibilisak a legkülönfélébb géptípusokon és operációs rendszereken. A fő gond viszont, hogy az elfogadott 852-es kódkiosztási szabvány ellenére nem képesek megegyezni a nemzeti karakterek elhelyezésében. Ugyanakkor a Bitstream mellett újabb

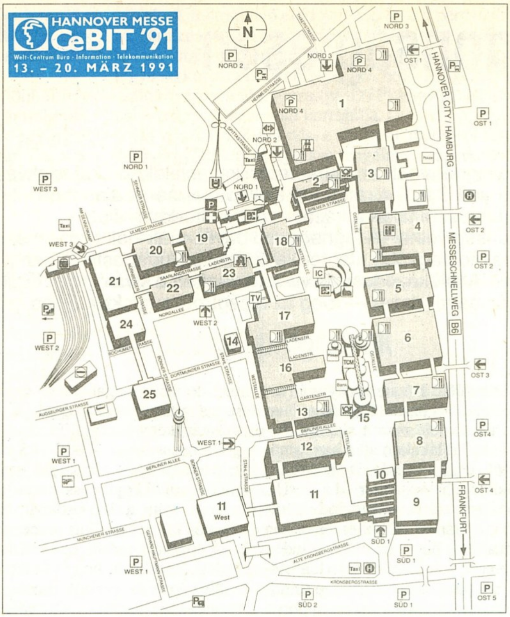
betűtechnológia jelent meg: az Adobe Type Manager-e, amelyet a 4.0 PageMaker mellett forgalmaznak. S ami rögtön szembetűnik: ez már szövegszerkesztőt is tartalmaz, tehát valóban komplett szedőműhelyt találunk az íróasztalon.

A képfeldolgozás is jeletősen egyszerűsödik. Megjelentek a képdigitalizáló és videokártyák. Ugyanakkor ezek a termékek egyre inkább eltérnek egymástól, nincsen szabványos vezérlési nyelvük, így legtöbbször csak a hozzájuk adott program használható. Ez különösen nagy gond a tajvani termékek esetében, ahol a hardveresen jó termék kriminálisan rossz szoftverrel párosulhat.

Megjelentek a televízió-vevő kártyák PC-hez. Ezek EGA vagy VGA monitoron a PAL és NTSC rendszerű programok kifogástalan vételét tudják megoldani. Legtöbbjük arra is képes, hogy a printscreen lenyomásával az éppen látott képet egy másik programmal feldolgozható formátumban elmentse. Erre annál nagyobb szükség lesz, minél inkább elterjednek az integrált multimédia rendszerek.

Az egyik legérdekesebb ilyen berendezést a holland TULIP számítógépgyártó cég mutatta be. Itt interaktív, PC-alapokon nyugvó információs rendszert fejlesztettek ki, amely képpel, hanggal és szöveggel informál. Az állóképet merevlemezen, a szöveget és a mozgóképet pedig optikai lemezen tárolták. Elég sok optikailemez-gyártó cég is jelentkezett termékeivel, az általuk kínált CD-ROM-oknak nagy sikerük volt. A CD-ROM-on kínált adatbázisok magánszemélyek számára — ritka kivételtől eltekintve — nem tartoznak a könnyen elérhető kategóriába, hiszen áruk 500 DEM és a csillagos ég között mozog. Az olvasóberendezések is mintha az adatbanki használatot támogatnák, hiszen akadt közöttük jó pár CD-ROM-ot befogadó „wurlitzer", azaz jukebox is... Csak itt a zenegép információt áraszt.

Ugyanakkor a multimédia végre helyet kapott az oktatásban, valamint az üzleti életben. Több cég kiállításán lehetett olyan UNIX-Xwindows vagy DOS-MS-Windows-alapú rendszereket látni, ahol az egyik ablakban televíziós képet vagy éppen több kontinensnyi távolságra lévő partnerünk fizimiskáját láthattuk élőben... Ehhez viszont egyre gyorsabb gépek szükségesek. Szinte nem is volt valamirevaló „vasgyártó" cég, amelyik ne mutatta volna be 386-os, esetleg 486-os alapú, jegyzetfüzet (notebook) méretű AT-ját.



Ezek winchester-kapacitása a 20 és a 200 megabájt között változott. A grafika természetesen VGA, és a beépített fax vagy fax/modem kombináció szinte mindennapos volt. Vajh' mit szólnak ehhez a Magyar Távközlési Vállalat engedélyezéssel foglalkozó és szabályzatokhoz láncolt munkatársai? Mert ezzel a technológiai lépéssel szerencsére végképpen kicsúszott a kezükből az irányítás. Olyannyira, hogy a hasonló közkedveltségnek örvendő, ám a mienknél jóval magasabb technikai színvonalon álló Deutsche Bundespost szakemberei már beismerték: ezt a játszmát végleg elvesztették. Különösen azután, hogy a Cebiten bemutatkozott Németország első teljesen magántőkés telefon-vállalkozása: a Mannesmann mobil rádiótelefon-rendszert épít ki az egész ország területén.

Az egyik nap kis csoport tiltakozott a kiállítás bejárata előtt a számítástechnikai kutatások és a számítástechnika azonnali betiltását követelve. A német zöldek egy kis csoportja egyenlőségjelet tett a környezetszennyezés, a katonai elektronika és a számítástechnika közé.

A Cebit kommunikációs pavilonjaiban postai és távközlési vállalatok kínálták ponttól pontig terjedő műholdas összeköttetéseiket. Ennek a technikának köszönhettük, hogy a CNN tudósítója szinte percenként tudott jelentkezni a lebombázott Bagdadból vagy a visszafoglalt Kuvaitból. Ezeket a könnyű berendezéseket ugyanis percek alatt lehet a világ bármely pontján telepíteni, és a telefontechnikához hasonló egyszerű választó eljárással felépíthető a kapcsolat akár egy másik hasonló berendezéssel, akár pedig a jelenlegi földi televíziós, rádiós, adatátviteli vagy éppen telefon- vagy telexhálózatok bármelyik előfizetőjével...

A műholdas technika most már nem olyan megfizethetetlenül drága. Két cég is bemutatta gépkocsik helymeghatározásának műholdas rendszerét. Itt a gépkocsikra egy speciális antennát és egy kisméretű ASCII terminált szerelnek. Ennek segítségével a központ méternyi pontossággal meg tudja határozni a gépkocsi helyzetét, ugyanakkor bármikor szöveges adatkapcsolatba tud lépni a vezetővel. Egy még jobban kezelhető változatot egy magáncég a rendőrségnek, a mentőknek, a taxisoknak és a szállítóvállalatoknak ajánlott.

A számítástechnika egyre inkább kommunikációs piaccá is válik. Az információs kavalkádban szerzett tapasztalatainkra későbbi számainkban, a konkrét témák feldolgozásakor folyamatosan visszatérünk.

Kis János

A GEM operációs rendszer IX.

Az adatrögzítők tehermentesítésére: adatkompatibilitás

Az eddigiekben a GEM rendszer külső megjelenésével foglalkoztunk, most pedig elkezdjük adatformátumának tárgyalását.

A Digital Research nemcsak a könnyű kezelhetőségre figyelt, amikor kifejlesztette a GEM felhasználói felületet, hanem szokásához híven — mint ahogy CP/M rendszerénél is — az adatkompatibilitásra is. Ők voltak az elsők, akik bevezették a redőnymenü-kezelést — ez a felhasználókat hamar meghódította —, de ők fejlesztettek ki elsőként olyan operációs rendszert a Macintoshra és az IBM PC-vel kompatibilis gépekre, valamint az Atari ST és TT gépcsaládra, amely az adatkompatibilitást is megteremtette a gépek között.

E két, említett kiváló tulajdonsága miatt hamar elterjedt a GEM a felhasználók táborában: manapság ki ne dolgozott volna vele, vagy legalábbis ki ne ismerné? A hazánkban népszerű Xerox Ventura Publisher DTP rendszer is GEM környezetet igényel.

Az eredetileg 1984-ben, Macintosh számítógépre íródott operációs rendszer külalakjában kissé eltér a gép vetélytársain futó GEM rendszerektől. Ennek legfőbb oka, hogy az Apple cég nem nézte jó szemmel, hogy a gépén futó operációs rendszer más számítógépeken is él, és beperelte a Digital Research-öt. A bíróság pedig a GEM rendszer néhány alapvető jellemzőjének elhagyására, illetve módosítására kötelezte a szoftverkészítő céget. Ezért van eltérés például a szemétkosár használatában és a kontrollpanel megjelenítésében. A lényeg azonban nem változott. A menürendszer ugyanaz maradt, és ami a legfőbb: az adatkompatibilitás sem változott. Mindhárom gépre kidolgozták a GEM Draw rajzoló- és a GEM Paint festőprogramot és a GEM Write szövegszerkesztőt. Az IBM PC-n és az Atari ST-n viszont a 1st Word Plus szövegszerkesztő rendszer terjedt el.

Most pedig vizsgáljuk meg, hogy mit jelent a gépek közötti adatkompatibilitás.

Magától értetődik, hogy a PC-re írt programok nem futnak a Macintoshon és az Atari ST-n, és ugyanígy fordítva: a Macintoshra vagy Atari ST-re írt programok sem futnak a PC-n. A programokat tekintve mindhárom gép inkompatibilis egymással. Ezért célszerű volt létrehozni egy olyan standard adatformátumot, amellyel — ha már az egyik gépre írt szoftver nem futtatható a többin — legalább a felhasználói programok által létrehozott adatokat a többi gépben is lehet kezelni.

Az adatformátum egyezőségén a következőket értem: a pixelgrafikus és a vektorgrafikus képformátum azonosságát, a fontkészletek megegyezését, a GEM Write programmal írt szövegállomány fájlformátumának kompatibilitását, a különböző driverek (printer driver, plotter driver) használatát bármely gépen.

Elsőként nézzük a GEM Paint programmal létrehozott IMG kiterjesztésű pixelgrafikus képformátumot. Ezt a formátumot támogatja az IBM PC-n futó Xerox Ventura Publisher és az Atari ST-n futó Calamus kiadványszerkesztő rendszer is, továbbá sok más olyan program, amely nem igényel GEM környezetet. Egy ilyen pixelgrafikus képfájl a fejrészből és a kép bittérképéből tevődik össze. A fejrész kétbájtos, azaz word-ös mezőkből áll, amelyek a képfájl megjelenítéséhez szükséges adatokat tartalmazzák (lásd a táblázatot). Az adatmezők számozása nullával kezdődik, amely az alkalmazott IMG formátum verziószámát adja. Az egyes számú mező a fejrészben található mezők számát, a kettes számú pedig a képpontokhoz tartozó grafikus szintek számát tárolja. Az ismétlődő minták tömörítésekor használt mintahosszúságot bájtokban mérve a 3. mezőn tárolja. A 4. és az 5. mező a pixelpont szélességét és magasságát adja meg ezred mm-ben, azaz mikronban — a képet létrehozó eszköz beállításai szerint. A 6. mező a kép szélességét, a 7. mező a képsorok számát tárolja pixelben megadva.

A képfájl fontos jellemzője, hogy a képernyő egy pixelének a színét hány bit határozza meg. Az alapszínek (piros, zöld, kék és szürke) esetében külön bittérkép adja a képinformációt, amit az IMG formátum bittérképenként egymás után tárol. A fejléc második mezőjéből olvashatjuk ki a bittérképek számát: ez nem más, mint a kép egy pixeléhez tartozó bitek száma. Abban az esetben, ha ennek értéke egy, akkor a fájl monokróm képet tárol, azaz a pixel színe csupán fehér vagy fekete lehet. A GEM alkalmazói felület által 16 színt lehet megjeleníteni, ezért a színes IMG típusú képek esetében a bittérképek száma (gyakorlatilag) maximum négy lehet. Ezek mindegyike egy-egy alapszín információját tartalmazza. A bittérképek sorrendje a következő színsorrendnek felel meg: piros, zöld, kék, szürke.

Elméletileg lehetőség van több bittérkép tárolására is, ugyanis a fájlformátum megengedi, csupán a színes grafikát is támogató rajzprogramok nem tudják kihasználni a lehetőséget. Az IBM PC-vel kompatibilis gépeken futó Ventura DTP rendszer, amely csupán szürke árnyalatokkal dolgozik, nem 4, hanem 8 bittérképet használ, ezért 256 különféle szürke színárnyalat megjelenítésére képes. Az IMG típusú fájl fejlécét közvetlenül a képadatok követik, amelyek soronként, ezen belül pedig bittérképenként tárolják a képinformációt. Az IMG fájl értelmezésekor a fejléc első mezőjéből feltétlenül ki kell olvasni a fejléc mezőinek számát, mert az előbbiekben felsoroltakon kívül az más mezőket is tartalmazhat, és ha ezeket nem vesszük figyelembe, akkor rossz helyről kiindulva kezdjük értelmezni a bittérképet.

A bittérképek tömörítési módjával a legközelebbi számban foglalkozunk.

Kovács P. Attila

Az IMG fájl fejlécének a felépítése

(Mezőszám és jelentése)
0 = Az alkalmazott IMG formátum verziószáma.
1 = A fejlécben található mezők száma.
2 = Egy pixelhez tartozó bitek száma (a bittérképek száma).
3 = Ismétlődő minták tömörítésekor alkalmazható mintahosszúság bájtokban.
4 = A képet létrehozó eszközben használt pixelszélesség mikronban.
5 = A képet létrehozó eszközben használt pixelmagasság mikronban.
6 = A kép szélessége pixelben.
7 = A kép magassága pixelben, azaz a képsorok száma.

A Medorg a magyar bajnok

E havi számunkban több helyütt is foglalkozunk a február végi—március eleji, Miskolcon megrendezett MikroCAD kiállítással. Úgy véljük, sokak érdeklődésére tarthatnak számot azok az információk is, amelyeket a MikroCAD keretén belül megrendezett I. Országos Számítástechnikai Programozói Bajnokságról kaptunk a Miskolci Egyetemen oktató, a versenyt szervező bizottság elnökétől, dr. Balla Lászlótól.

A 24 órás, nonstop versenyen maximálisan 3 fős csapatok vehettek részt, korlátozva volt a részt vevő csapatok száma is. Ezt 24-ben maximálták, de végül is a borsos nevezési díj — 40 000 forint — sokakat visszatartott, így mindössze 9 csapat szállt ringbe a büszke magyar bajnoki címért.

A versenyre a következő csapatok neveztek: Medorg Rt. (Palkó Gábor, Börzsei Zoltán, Vári László), USERLAND (Bolgár Gábor, Zsakó János, Lénárt Zoltán), MosaiC (Bíró Attila, Jedlicska Zoltán, Reményi Piroska), Szinva (Földi Péter, Zahorán Péter, Tisza Miklós), ESCAPE (Szabó István, Balog L. Csaba, Pivarnyik Attila), Alfadat Team (Brunner József, Láng János, Várnagy Péter), Kosir Attila Team (Kosir Attila, Haáz Győző, Jászberényi Márk), VSZM Közösségi Ház Micro Magic (Soós Gábor, Ladányi Péter, Nagy Károly), Földes (István György, Fedorcsák Péter, Zámborszky Ferenc).

A zsűri tagjainak sorában a magyar szakemberek mellett Svédországból érkezett vendégek is helyet kaptak. Ezt egyrészt az indokolta, hogy a versenyforma világbajnokságát ősszel Svédországban rendezik, másrészt az is, hogy a versenyt a svédek saját belső projektjeik szakmai támogatásául találták ki, a megoldandó feladat szorosan kapcsolódik a negyedik generációs adatbázis-kezelőkkel kapcsolatos svéd kutatásokhoz. A zsűri elnöki tisztét dr. Dömölky Bálint töltötte be, a tagok között Lennart Aspenryd és Bengt Björnekärr képviselte a szervező Svédországot, rajtuk kívül Köves Péterre, dr. Koch Péterre és dr. Vajk Istvánra hárult a versenyprogramok elbírálásának korántsem egyszerű feladata.

A verseny szponzoraiul a Miskolci Egyetemen kívül jónevű cégeket — IBM, Microsoft, Packard Bell, Siemens, Számalk — sikerült megnyerniük a szervezőknek, nevüket már csak azért is érdemes megemlíteni, hátha éppen ez segít abban, hogy jövőre ismét ott legyenek a támogatók sorában.

A versenyfeladatot Bengt Björnekärr és Lennart Aspenryd svéd számítástechnikai szakemberek készítették a svédországi programozói világbajnokság követelményeinek megfelelően.

A feladat egy adatkezelést, megjelenítést, gyűjtést megvalósító program megírása volt. A rugalmas és teljes körű rendszernek elegáns képernyős megjelenítéssel, áttekinthető menürendszerrel, ellenőrzött bevitellel és minden lehetséges esetre „help"-nyújtással kellett rendelkeznie. A feladat leírása nem tartalmazott teljes körű rendszertervet, csak ennek bizonyos elemeit: fájlok, rekordstruktúra, néhány listakép, a megvalósítandó folyamat vázlatos leírása. A feladat tehát tulajdonképpen egy konkrét információs rendszer teljes körű, intelligens, felhasználóbarát, gyors kifejlesztése volt.

Konkrétan egy kereskedő cég árukészletének, szállítóinak nyilvántartását kellett elkészíteni. Figyelnie kellett a rendszernek az árukészlet adott határ alá csökkenését újrarendelés céljából. A rendszernek tartalmaznia kellett a cég vevőkörének, eladásainak, eladóinak nyilvántartását, az egyes eladók üzletszerzési tevékenységének nyomkövetését, ennek alapján a jutalékszámítást, valamint az eladások előrejelzését a vevők múlt évi rendelései és a mostani vásárlási ajánlatai alapján. Az adatok lekérdezhetőségét minél több szempont szerint lehetővé kellett tenni. A listák egy része arra szolgált, hogy az ügynököknek segítséget nyújtson tevékenységük sikeréhez. Figyelnie kellett a rendszernek arra, hogy kik azok a rendszeres vevők, akiknek engedmény adható, kinek kell karácsonyi üdvözlőlapot küldeni, kinek kell levelet küldeni vagy telefonálni a következő héten a vevők figyelmének felkeltése céljából. Ez utóbbi követelmény az előző évi vásárlás alapján prognosztizálható árucikkeket vette alapul. Gondoskodni kellett továbbá az adatok illetéktelen hozzáféréstől való védelméről is.

A verseny első díját (a csapat benevezése az NJSZT által a svédországi világbajnokságra + 100 000 Ft költséghozzájárulás a verseny szponzoraitól a kinntartózkodásukhoz) a Medorg Rt. csapata nyerte. IBM AT-alapú NOVELL lokális hálózattal, lézernyomtatóval, mátrixnyomtatóval dolgoztak. Szoftverként MAGIC II. adatbázis-kezelőt, StatGraf-ot, Turbo Pascal 6.0-t, Xerox Ventura Publishert és WordPerfect szövegszerkesztőt használtak.

Második a USERLAND Gmk. csapata lett. Ők 3 AT 386-os gépen dolgoztak 2 nyomtatóval és egy Laptop 386-tal. Clipper 5.0 nyelven programoztak. A szponzorok jóvoltából ők is utazhatnak a svédországi világbajnokságra.

A MosaiC csapat lett a harmadik. Feladatuk megoldásához 2 IBM AT 286-os gépet, 1 IBM AT 386-ot, VGA monitorokat, 1 EPSON FX 1050 nyomtatót, egeret, kivetítőt használtak. Turbo-C 2.0-ban programoztak, MosaiC fejlesztőrendszerrel, StatGraf 2.0, AutoCad 2.6-ot is felhasználtak. Jutalmuk Packard Bell 286 számítógép lett.

A zsűri különdíját a Szinva csapata nyerte el, jutalmuk 5000 DEM értékben jogtiszta Microsoft-szoftver. Ők VGA kártyás AT-t használtak, 5.5-ös Pascalban, valamint Clipper 5.0-ban programoztak. Rajtuk kívül eredményes volt még az Escape feladatmegoldása.

A feladat jellegéből adódóan nem volt szükség a számítógép operációs rendszerének teljes mélységig való ismeretére, ha a versenyzők megfelelő adatbázis-kezelő rendszert választottak a megoldáshoz.

Útvesztő vagy vezető az úton?

A problémától a programig

15 évvel ezelőtt a programozást a Fortran és hozzá hasonló nyelvek, a lyukkártyás technika jelentette. A feladatok megoldásakor nem volt szükség az akkori legújabb szoftvertechnológia ismeretére. A nyolcvanas évek kezdetét a különböző informatikai tanfolyamok, a PC-k megjelenése, az assembler, a PLM, a dBase, a Turbo Pascal elterjedése jellemezte, de ez sem hozott minőségi változást ebből a szempontból. Az alábbiakban — a Toolbox c. magazin nyomán — egy olyan cikket adunk közre, amely problémafelvetését tekintve a Modula-sorozat megállapításaival is vitatkozik.

A fordulat a nyolcvanas évek vége felé következett be, amikor a PC-k térhódítása miatt számos újságban jelentek meg cikkek; ezekben egyre gyakrabban szerepeltek olyan kifejezések, mint „strukturált programozás", „információrejtés", „moduláris koncepció" vagy „Jackson-módszer". Mindenhol ezekre esküdtek, ezeket emlegették a szoftverfejlesztés csodaszereiként. Korábban a programozók nagy része abban a hitben élt, hogy minden feladatot elsősorban a józan paraszti ész alapján lehet megoldani. (Bármilyen hangzatos elnevezésű programtechnikák kerültek reflektorfénybe, ez azért mindmáig igaz maradt.)

Bosszantó volt követni a Basic-pártiak és a Pascal-hívők állandósult vitatkozását, öndicséretét, a másik becsmérlését. Érthető ezek után, hogy számos kezdő, de jó néhány gyakorlott programfejlesztő is elbizonytalanodott. Dunát lehetett rekeszteni nemrég még a spagettiszerűen, szépen sorban egymás után leírt utasításokból álló programokkal, most meg már itt van az OOP (objektumorientált programozás) — ez a sláger!

Minden módszernek megvan természetesen a maga előnye. Felmerülhet azonban itt is a kérdés: mi volt előbb, a tyúk vagy a tojás. Az OOP már azt jelentené, hogy ezentúl csak szuperprogramokat tudunk írni? Kimondhatjuk, hogy egy jól átgondolt program attól, hogy Basic nyelven írjuk, máris érdektelen? Nézzük meg most egy kicsit a szokásostól eltérően gondolkodva, különböző szempontok szerint azt az utat, amely a probléma megfogalmazódásától a program elkészítéséig vezet!

Válasszunk egy rövid segédprogramot demonstrációs célra! Ez DOS alatt igen hasznos, ASC névre hallgat, az utasítássorban paraméterként hexa jeleket kap. Ezeket ASCII kódra alakítja, majd az így kapott jeleket a megadott kiviteli eszközre küldi. A hívás szintaxisa ilyen egyszerű:

ASC<p1><p2>...<pn>

A fenti sorban a <p.> a paraméterként beadott kétszámjegyű hexa számot jelöli. Ha több paraméter követi egymást, ezeket egy-egy szóköz választja el. Az alábbi hívások helyesek például szintaktikailag:

ASC 41 42 43
ASC 08
ASC 0C > PRN:

Az első példasor hatására a program az „ABC" szöveget fogja kiadni, a második egy hangjelet eredményez, a harmadikkal pedig lapot dobunk a sornyomtatón. A sornyomtatóra különböző jelsorozatokat küldve (pl. ASC 1B...) DOS-ból vagy batch programból változtathatunk írásfajtákat vagy akár sortávolságot is. Könnyen előállíthatunk COM állományokat is.

A pontos feladat igényes megoldása

A feladatot tehát megfogalmaztuk, most jön az érdekesebb kérdés — készítsük el a futtatható programot!

Mielőtt leírnánk az első programsort, ajánlatos legalább nagy vonalakban felvázolni a tervezett megoldást. Javasolhatjuk a következő lépéseket: a paramétereket leválasztjuk az utasítássorból, dekódoljuk a hexa jeleket, kiadjuk az ASCII kódot. Ezt a szerkezetet az alábbi Pascal-szerű utasításokkal mutatjuk be:

```
FOR minden paraméter
leválaszt paramétert
FOR minden jelre a paraméterben
IF érvénytelen jel THEN
Hibakilépés
Jelet hexa számmá alakítani
END FOR
ASCII jel kiadása
END FOR
```

Az ASC program első verziója lineárisan dolgozik. A paraméterek leválasztása a „ParamStr" belső Turbo függvénnyel történik. Ekkor csak néhány ciklus szükséges a paraméterek beolvasásához, dekódolásához és a megfelelő ASCII jelek kiadásához. A megoldás egyszerűsége ellenére is maradéktalanul kielégíti a megfogalmazott igényeket. (1. lista)

A probléma megfogalmazásakor azonban nem emeltük ki azt, hogy a feladat bonyolultsága terjedelmessé fogja tenni magát a programot. Így a programok nehezen érthetővé és lassúvá válhatnak. Ezenkívül még az összes utasítást is egyenként kell kódolnunk, nem lehetséges az egyes programrészek átvétele más programokba.

A hibákból illik tanulni, az adott konvenciókat pedig illik betartani. Térjünk vissza az eredeti feladathoz! Ha áttanulmányozzuk az 1. listát, már azonnal adódnak javítási ötletek: a hibás input kilépési pontjára többször hivatkozunk, itt procedúrát kellene alkalmazni. A hexa jelek érvényes hexa értékekké konvertálása is többször ismétlődik. Ha itt függvénnyé alakítjuk a megfelelő programsorokat, akkor csak meg kell hívnunk ezt a függvényt, és későbbi projektek programjaiban is használhatjuk majd. Ha ezt a két dolgot megtesszük, lerövidül maga a főprogram, így jobban áttekinthetővé válik. A procedurális programozás fent vázolt technikája kiválóan megvalósítható Pascalban. (2. lista) Mindez igaz ugyan és nagyon szépen hangzik, de az így kialakított lista hosszabb lesz az előre kialakított procedúrák miatt.

A más programokból történő átvételnek is van hátulütője: minden átfutáskor újra kell fordítani a procedúrákat. További problémákat okozhat az is, hogy ezek a procedúrák forrásnyelven állnak rendelkezésre. Ha könyvtári rutinok lennének, egyszerűen csak meg

kellene hívni ezeket. Így azonban a programozók a forráskód láttán azonnal felbuzdulnak, mindenki csak egy kicsit jobbá, szebbé teszi, ennek a következményeit pedig nem nehéz elképzelni — különösen nagyobb projektek esetén, ahol több programozó dolgozik együtt, és mindenki igazán csak jót akar a módosításokkal.

Pascal helyett: Modula?

Az ilyesfajta hibák a Pascal nyelvnél gyakoriak, ezeket védi ki a Modula nyelv. Ez utóbbi ugyanis megengedi, hogy az egyes modulokat külön-külön fordítsuk, illetve implementáljuk. A Borland cég valami hasonlót nyújt a Turbo Pascal 4.0 feletti verziókkal: az egységek koncepcióját. Itt elkülönítve fejleszthetjük ki az egyes procedúrákat, függvényeket, ezeket fordítás után könyvtárakból hívhatjuk. Így megoldódik a programvédelem is. A 3. listán követhetjük az ASC probléma egy egységének kialakítását. Külön fordítjuk a „Lib"-et, TPU fájlként szerkesztjük hozzá a főprogramhoz. A moduláris programozás néhány utasításra és procedúrahívásra redukálja programunkat. A fejlesztőt nem terheli felesleges forrásszövegekkel, teljesen a munkájára tud koncentrálni.

Itt kellene összefoglalni tapasztalatainkat, de ezek egyáltalán nem egyértelműek, komoly vitákra adhatnak okot. Érdekes mindenesetre, hogy a különböző módszerekhez tartozó EXE állományok milyen méretűek:

Lineáris programozásnál: 4976 bájt

Procedurális programozásnál: 5063 bájt

Moduláris programozásnál: 5209 bájt

Ha így nézzük, egyértelműen az 1. listán szereplő lineáris program lenne a nyerő. Aki még tovább bogarássza ezt, hogy néhány bájtot megtakarítson, gondoljon arra, hogy ugyanez assemblerben írva mindössze 200 bájt lenne! Igazi MEGOLDÁS tehát NINCS. Csak olyan kategóriáknak van értelmük, mint „legelegánsabb" vagy „legegyszerűbb". Természetesen ugyanígy megírható a „legbonyolultabb" vagy a „legkörülményesebb" program is. Minden egyes feladat tartalmazza a maga sajátos követelményrendszerét. Válasszuk mindig azt a módot, amely nekünk személy szerint a legtesthezállóbbnak tűnik! Még akkor is, ha esetleg egyes kollégák sajnálkozva mosolyognak programunk láttán. (Túl egyszerű, illetve nem elég elegáns szerintük.)

V. M.

MODULA-2

Hogyan legyen új a régi?

A modulstruktúra lehetőséget nyújt újrafelhasználható szoftverkomponensek fejlesztésére. A újrafelhasználható komponensek elve egyszerű bölcsességet követ: könnyebb egy már bevált programot használni, mint egy újat teremteni a semmiből. Az efféle fejlesztés általános problémája a különböző alkalmazásokhoz való illesztésük szokott lenni. Például egyik esetben az kívánatos, hogy a modulnak legyen meg a saját hibakezelése, míg ugyanennek a modulnak egy másik felhasználásakor a modul kliense privát hibakezelést kíván. És így tovább...

Eljárásparaméterek — eljárásváltozók

A Modula-2 orvossága ezekre a problémákra az eljárásváltozók bevezetése, ami a Pascal eljárás-paramétereinek általánosítása. Az eljárásváltozók olyan változók, amelyekhez értékként eljárásokat adhatunk meg. Egy eljárásváltozó kijelölése a hozzá tartozó eljárás hívását eredményezi.

Az eljárásváltozók lehetővé teszik a kliens modulok számára saját eljárásaiknak a könyvtári modulokhoz való rendelését. Például egy kliens a saját hibakezelő eljárását rendelheti egy könyvtári modul megfelelő változójához, és így — hiba esetén —, amikor a könyvtár meghívja a belső hiba eljárásváltozóját, akkor a kliens hibakezelője hajtódik végre.

Az eljárásváltozóknak egy másik felhasználása a könyvtárak automatikus inicializálása és lezárása. Ez a könyvtármodul számára lehetővé teszi saját erőforrásainak indítását és leállítását a kliens moduloktól függetlenül. Például egy hálózatkezelő modul inicializáló eljárása a kapcsolat létrehozásához szükséges hardverműveleteket végzi el, míg a lezáráskor a modul automatikusan lebontja a kiépített kapcsolatokat. A hálózatkezelő ezeket az eljárásokat a programtöltőnek adja át, aztán a modul betöltésekor, illetve felszabadításakor hajtódnak ezek végre. A kliens modulok nem tudnak ezekről a részletekről; csupán annyit, hogy a modul egy hálózati csatlakozást bocsát a rendelkezésükre.

A 3. lista az eljárásváltozók használatát mutatja hibakezelésre és inicializáló/lezáró eljárásokra egy hálózatkezelő modulon belül.

Gépközeliség és hordozhatóság

Mivel az alacsony szintű rendszerelemek könyvtármodulokban találhatók meg, ezért a Modula-2 tartalmaz a gépszintű programozáshoz szükséges elemeket is. Ezek az elemek természetüknél fogva gépfüggőek, ezért a Modula-2 aktuális implementációjától is függenek, így a hordozható programokban célszerű elkerülni őket. A gépszintű programrészeket általában néhány alacsony szintű modulba foglalják, elválasztva őket a hordozható részektől. A rendszer transzportálása esetén elegendő ezek módosítása a rendszer megbontása nélkül.

Az előre definiált BITSET típus olyan halmaz, amely egy gépi szóban ábrázolható. A bithalmazok lehetőséget teremtenek szavas egységek bitenkénti manipulációjára.

Az előre definiált SYSTEM modul a WORD, ADDRESS, SIZE, TSIZE és ADR rendszerfüggő azonosítókat exportálja. Amikor a WORD típust paramétertípusként használjuk, akkor az összes, egy gépi szóban ábrázolt objektummal kompatibilis. Ha a WORD típust nyílt tömbparaméter bázistípusaként adjuk meg (ARRAY OF WORD), akkor a paraméter bármilyen méretű és típusú változóval kompatibilis. A WORD típus lehetővé teszi tetszőleges változókon manipuláló, általános célú rutinok írását.

Az ADDRESS típus a CARDIONAL (természetes szám) típussal és minden mutatóval kompatibilis. Az aritmetikai operátorok az ADDRESS típusra is értelmezve vannak, így cím- és mutatószámítások elvégzésének nincs akadálya. Ez fontos eszköz tárkezelők megvalósításakor; különösen azért, mivel a Modula-2 a szabványos NEW és DISPOSE eljárásokat az alacsony szintű ALLOCATE és DEALLOCATE eljárásokra fordítja le. Ezeket az eljárásokat általában egy szabvány tárkezelő modul tartalmazza, de más módokon is definiálhatók — ha eltérő tárkezelés kívánatos. (Mivel a Modula-2 önmagában füg-

getlen az operációs rendszertől, ezért szükség lehet ezen eljárások Modula-2-ben való megvalósítására, ha az adott operációs rendszer nem támogatja a tárkezelési funkciókat.)

A SIZE (x) és TSIZE (x) függvények változók és típusok méretét adják vissza, míg az ADR (x) függvény csak a változók címét.

Változókat fix memóriacímre is deklarálhatunk. Ezeket például memóriába ágyazott regiszterek kezelésére lehet jól használni.

A szigorú típusellenőrzés megkerülhető azáltal, hogy a típus nevét mint típuskonverziós függvényt vesszük figyelembe, ami a paraméterét a függvény azonosító típusára konvertálja. A típuskonverzió csak azonos méretű változók esetében működik.

A kódeljárások teszik a Modula-2 programok számára természetessé a hardver vagy az alacsony szintű rendszer közvetlen elérését. Egy kódeljárás egy eljárásfejlécből és az utána következő CODE kulcsszóból áll, amelyet egy sorozat gépi utasítás követ; ezek az utasítások minden hivatkozáskor „inline" kerülnek a kódba. A kódeljárások használatát célszerű az alacsony szintű modulokra szorítani.

A 4. lista a Modula-2 gépszintű eszközeit szemlélteti.

Multiprocesszálás korutinokkal

A legtöbb modern rendszerprogramozási nyelv fel van készülve multiprocesszálásra. A Modula-1 például processzekkel és szemaforral, az ADA taszkokkal és randevúval, a Concurrent Pascal pedig monitorral. A Modula-2 lemond ezekről a magas szintű multiprocesszálási modellekről: helyette az egyszerűbb és alacsony szintű korutin fogalmat vezette be, mely a definíciónak megfelelően működik a multiprocesszálás kapcsán.

A korutinok olyan eljárások, amelyek egymástól függetlenül, de nem konkurens módon hajtódnak végre. A korutinok egy csoportjában csak egyetlen korutin hajtódik végre, a többi nem aktív. A korutinok egymás meghívásával kommunikálnak, egy korutin meghívásakor a hívó korutin futása felfüggesztődik, a hívott korutin válik aktívvá. A futását ez azon a ponton folytatja, ahol legutóbb felfüggesztődött.

A korutinnal kapcsolatos elemeket a SYSTEM modul exportálja. Ezek a NEWPROCESS, A TRANSFER és az IONTRANSFER eljárások és a PROCESS típus.

A NEWPROCESS (Procedure, Adr, Size, Corutine) eljárás új korutint határoz meg. Ennek Procedure paramétere azt az eljárást jelöli, amelyiket a korutinnak végre kell hajtania. Az Adr és Size paraméterek a memória egy területét jelölik ki a korutin számára a memóriában. Az újonnan létrehozott korutint a PROCESS típusú Corutine paraméter tartalmazza.

A korutinhívásokat a TRANSFER (Old, New) eljárással valósíthatjuk meg. Az eljárás felfüggeszti az éppen futó korutint, ezt az Old változóhoz rendeli, majd a New paraméterhez rendelt korutin futását a korábbi felfüggesztésétől folytatja.

Az IOTRANSFER (Old, New, IntVect) eljárás lehetőséget nyújt megszakítások kezelésére korutin segítségével.Az IOTRANSFER egy korutinhívást valósít meg Oldról Newra, azonban ha az IntVectnek megfelelő megszakítás következik be, a vezérlés automatikusan visszakerül a megszakítás-kezelő rutinra. Az 5. lista a korutinokkal megvalósítható megszakítás-kezelésre mutat példát.

A korutinok sok gyakori multiprocesszálási alkalmazás számára megfelelőek, és ha magasabb szintű processzütemezésre van szükség (például szemaforozás vagy üzenetküldés), akkor ezek is a korutinok segítségével valósíthatók meg, és könyvtárakba szervezhetők. Így a Modula-2-vel tetszőleges ütemezésű (akár ütemezés nélküli) multiprocesszálás valósítható meg.

„Extrák"

A Modula-2 több, mint egy modulstruktúrával, alacsony szintű programozással és multiprocesszálással bővített Pascal. Ezeken felül a következő változtatásokkal büszkélkedhet a Pascalhoz képest, amelyek egyszerűsítik a programozást és növelik a programok olvashatóságát és hatásfokát (lásd a 6. listát).

— Minden strukturált utasítás egy explicit záró utasítással végződik: az UNTIL a REPEAT-é, míg az END minden más utasítás lezárása. Ez feleslegessé teszi a Pascal-beli összetett utasításokat (BEGIN...END).

— Mindenütt, ahol a Pascalban állandók szerepeltek, a Modula-2 állandó kifejezéseket enged meg (TYPE a = ARRAY [0..n -1] OF CHAR).

— A CASE utasításnak részintervallum is szerepelhet a választási címkéiben; így nem szükséges minden elemet külön felsorolni. A CASE utasításnak van egy opcionális ELSE ága.

— A variáns rekordok szintaktikája megegyezik a CASE struktúráéval.

— A függvények visszatérő értéke tetszőleges típusú lehet.

— Az eljárások úgynevezett nyitott tömböket deklarálhatnak paraméterként, amelyek tetszőleges hosszúságú tömböket fogadnak el paraméternek.

— A logikai operátorok kiértékelésekor a második operátor kiértékelése nem vesz időt, ha a kifejezés értéke az első operandusból is meghatározható. Ez sok esetben leegyszerűsíti a programozást.

— A LOOP/END struktúra lehetővé teszi belső megszakítású ciklusok létrehozását elegáns és érthető módon.

— A FOR ciklusnak van egy opcionális BY része is, amely a lépési érték meghatározására szolgál.

— A Modula-2-ben minden karakter szignifikáns, nemcsak az első n darab, így az azonosítók „beszédesek" lehetnek anélkül, hogy véletlenül megegyezne az értelmezésük. A nagybetűvel írandó kulcsszavak kiemelkednek a programlistából, és ezért az olvashatóságot és az érthetőséget növelik. A Modula-2 megkülönbözteti a kis- és a nagybetűket, így aa, aA, Aa, AA négy különböző azonosító.

Villányi László

Az eddigiek mint bevezetés elég alapot adnak ahhoz, hogy lassan a konkrét megvalósítások részletezésére, elemzésére is sor kerüljön. A következő hónapban az egyik Modula-2 implementációról és annak sajátosságairól esik majd szó — összehasonlítva ezt a rendszert a legjelentősebb PC-s fordítókkal. Tehát a sorozat mélyebb témákhoz érkezik, s ez előtt — a „nyelvi" villongásokat elkerülendő — hadd segítsen egy idézet fölkelteni azoknak az olvasóinknak az érdeklődését is, akik jól megvoltak eddig a Modula-2 nélkül.

„Sokan elismerik, hogy elengedhetetlen egy olyan jelölésmód használata, amely tisztán és érthetően fejezi ki az elgondolásokat, továbbá kiemeli a rendezett struktúrák szükségességét. Legalább ennyire elterjedt azonban az a nézet is, mely szerint az ár ezért az érthetőségért és strukturáltságért megfizethetetlenül nagy, és a magas szintű nyelvek használata megengedhető a tantermekben, de nem az iparban uralkodó versenyben. Természetesen ez a állítás nem maradt meg a hiedelmek kétes rangján, hiszen bebizonyítható volt. ... A teszteredmények — noha meggyőzőek és igazak, — egy félreértésen alapultak, amelyet az okozott, hogy sok ember nem tudott különbséget tenni egy programozási nyelv és implementációja között. A programozási nyelv egy tömör és formális definíción alapuló absztrakciós készlet, amely alkalmas algoritmusok és adattípusok kifejezésére. Az implementáció viszont mechanizmus, amely a formalizmussal kifejezett programok értelmezését szolgálja. Sajnos úgy érzem, hogy mind a mai napig sok szakember, tanárok és írók is, nem ismerik fel ezt az alapvető összefüggést." (Wirth professzor)

Amikor a folyam(at)ok visszafelé folynak

Clipper '87 — Decompiler V1.0-H

A szerző munkájának kiindulási tárgya a Clipper '87 nyári verziója, melyet a Microsoft cég 5.0-s C rendszerében fejlesztettek ki, így magán viseli a nyelv összes ismertetőjelét. Ehhez a szoftverhez azután Turbo Pascal 5.5-ben megírt egy decompilert, melynek a CLDEC87-Hungary programnevet adta.

A fordítóprogramok működésének elméletében jártas olvasó nyugodtan kihagyhatja a cikk elejét, ezt a rövid ismertetést, amelyben megkísérlem összefoglalni mindazt, amit e fogalom takar. Mankóként azért idézem az Oxford számítástechnikai szótár idevonatkozó cikkelyét:

DECOMPILER: „visszafordító program: program, amely megkísérli, hogy egy fordítóprogram eredményén elvégezze ugyanazt a műveletet, amit egy disassembler egy assemblálás eredményén végez el, azaz a gépi kódot valami olyasfélére fordítsa vissza, ami a forrásnyelvű programra emlékeztethet."

Mit, miből, miként vissza?

A compilerek/fordítóprogramok egy adott nyelv szabályai szerint megírt programot alakítanak át a számítógép hardvere számára értelmezhető utasítások sorozatává.

A fordítóprogramok vagy Assembly szintű programot, vagy abszolút bináris kódot, vagy áthelyezhető formátumú (például MS-DOS operációs rendszerben .EXE kiterjesztésű) programot állíthatnak elő. Egyes fordítóprogramok nem abszolút kódot, hanem egy átmeneti, úgynevezett pszeudokódot (P-kódot) generálnak, amelyet egy gyors belső interpreter dolgoz fel és hajt végre.

A fordítóprogramoknak általában négy feladatot kell elvégezniük:

— LEXIKÁLIS ELEMZÉS
— SZINTAKTIKAI ELEMZÉS
— SZEMANTIKAI ELEMZÉS
— KÓDGENERÁLÁS

A lexikális elemzés során a lefordítandó programot a fordítóprogram a legkisebb értelemhordozó egységre, szavakra bontja, és megvizsgálja, hogy ezek beletartoznak-e az adott nyelv alapkarakter-készletébe.

A szintaktikai elemzés a szintaktikai szabályok betartását vizsgálja: a program elemei a nyelv jelkészleteiből a szintaktikai szabályok alapján leképezhetők-e.

A szemantikai elmzés a program logikai hibáira derít fényt, például a hibás ciklusszervezésre stb.

A kódgenerálás során a fordítóprogram a fordítás során létrehozott táblázatok és belső alakok alapján állítja elő a végrehajtható formátumú programot.

Érdemes megjegyezni, hogy a fordítóprogramok többsége a fordítási folyamat alatt valamilyen szempontból optimalizál is — úgy, hogy a szemantika megsértése nélkül hatékonyabb kódot generál: a helyfoglalás, a futási idő stb. szempontjából.

Ezeknek a folyamatoknak ismeretében a decompilernek a következő feladatokat kell megoldania:

— Az OUTPUT fájl alapján fel kell ismernie a program (CODE), adat (DATA), konstans (CONST) szegmenseit, illetve az adott rendszerhez kapcsolódó egyéb szegmenseknek, például a veremnek (STACK) a pontos elhelyezkedését.

— A feltérképezett szerkezet alapján azonosítani kell a forrásprogram elemi egységeit, valamint az azokból képzett összetett szerkezeteket (például függvényhívásokat).

— Újra fel kell építeni a változók, tömbök, függvények, konstansok, (címkék) táblázatait.

— A forrásnyelv képzési szabályai szerint rekonstruálni kell az eredeti programot.

A felvázolt tevékenységek az alkalmazott fordítóprogramok függvényében módosulhatnak.

Lássuk a medvét!

A vizsgált verzióban a belső C STANDARD függvények bájthatárra vannak igazítva. A szerkesztéskor kapott MAP fájl alapján a belső interpreter függvényei láthatók, valamint a C környezet szegmensei.

Az Alaplap mostani számának mágneslemez mellékletén közreadott lista a COMMENT.PRG program MAP fájlját tartalmazza.

Most már előttünk a „gerinc", amelyre a többi rész épül. A kódgeneráló működését kell kiismernünk. Ahhoz, hogy ezt megtehessük, a Clipper nyelv formális leírását kell megalkotni. Ezt BNF formában határoztam meg. (A BNF a Backus-Naur Form betűszava, amellyel az ALGOL 60 nyelvet definiálták az „ALGOL 60 jelentésben" . Ebben a nyelv szintaxisát BNF alakban adták meg.)

A nyelv formális leírásával generálni tudtam a nyelv helyes mondatainak véges halmazát, amellyel a kódgeneráló tevékenységét térképeztem fel. Nehezebb probléma volt a generált adat- és struktúrahalmaz logikus elemzése, amely egy rafinált elemző megírását követelte meg.

Ha sikerült a forrás- és célnyelvi megfeleltetés szabályaira rájönni, már csak írni kellett egy programot, amely összekapcsolja a programok kód-,konstans- és adatterületeit. Ezt a feladatot a LINKER nehezíti meg, amely a szabványos OBJECT formátumú modulokat a globális és lokális nevek és egyéb információk alapján szerkeszti össze az egyes modulok hivatkozási címeinek figyelembevételével.

Az összeszerkesztés eredményeként előállt a végrehajtható formátumú, EXE (és OVL) kiterjesztésű program.

Ez a programfájl tartalmazza a DOS számára szükséges relokációs adatokat, amelyek alapján a relokálandó címeket felülírja az aktuális szegmens:offset értékkel.

A CLDEC87-H főbb moduljai: FULLMAP, DECOMP, CONTROL.

A FULLMAP modul

Ennek a modulnak az a feladata, hogy az EXE és OVL (overlay) programok feltérképezését elvégezze, és az alábbi elemek kezdő- és végcímeit meghatározza:

— Clipper nyelvű programszegmensek

— ASSEMBLY és C nyelvű bináris szegmensek

— RUN-TIME szegmensek

— OBJECT modulok

— SYMBOL (szimbólum) tábla elemei

A létrehozott MAP fájl alapján szétdarabolja az EXE és OVL fájlt XXXXXBIN.XXX és XXXXXOVL.XXX bináris fájlokká. A modul az ellenőrző fázisban az újrafordított programszegmenseknek XXXXXCTR.XXX fájlnevet ad.

Az OBJECT szerkezet (a szimbólumtábla információi) alapján az OBJECT.XXX és OBJLNK.XXX fájlok készülnek el. Az OBJECT.XXX fájlban a forrásprogramban előforduló változó-, eljárás-, függvények nevei első előfordulásuk sorrendjében szerepelnek. Az ellenőrző fázisban ugyanezek CHK OBJ.XXX nevet kapnak. Az OBJ LNK.XXX fájl az object modulba szerkesztett szegmensneveket tartalmazza.

Eközben létrejön egy PROGRAM.FNS (exe illetve OVERLAY.FNS (ovl) kiterjesztésű fájl, amely a visszafordítási eljárást vezérli. Tartalmazza a bináris fájlok neveit (XXXXBIN.XXX szimbólum nevét, például forrás eljárás- vagy függvénynév, a hozzá tartozó szimbólumtábla neve: OBJECT.XXX).

A DECOMP modul

Ez a modul alakítja át a P-kódú Clipper-szegmenseket forrás formátumúvá. A visszafordítás folyamatát a PROGRAM/OVERLAY.FNS fájlban lévő információk vezérlik. Lehetőség van kötegelt, illetve egyenkénti visszafordításra is. A változó- és procedúraneveket az OBJECT.XXX fájlból olvasom ki. Nézzünk egy egyszerű példát:

Forráskód:
(1) string=„Clipper"
(2) ? string

P-kód:
(1) 64 01 00 97 07 43 4C 49 50 50 45 52 00 D8 02 00 89
(2) 64 02 00 8D 9E 02 00 8C
OBJECT.1:

PRGNÉV (bármely név legfeljebb 10 karakter hosszú lehet)
STRING
(EOF)

Ha -l opció nélkül fordítunk, a compiler minden sor elé három bájtból álló sorszámot tesz. A hexadecimális 64 mutatja az interpreternek, hogy sorszám következik. Ez a DEBUGGER számára, illetve a futás közbeni hibák azonosításához szükséges információ. A következő 2 bájt LO-HI sorrendben mutatja a sorszámot.

A 97 azt jelenti, hogy sztring következik. A következő bájt a hosszt, a szöveg utáni 00 bájt a C nyelv szeriti sztringvéget jelzi.

A D8 a változó értékadását jelenti, a 0200 a 2.változóra mutat a szimbólumtáblában. Minden OBJECT modulnak saját szimbólumtáblája van, a legelső név az object nevét adja, a következők a program azonosítóit mutatják.

A 89-es bájt az értékadó operátor.

A 8D jelöli a ? kezdetét, 9E a hivatkozást a 0200 2. változó értékére, a 8C a ? -et zárja le.

Sajnos a Clipper 87 nyári verziójának több továbbfejlesztett változata van. Így bizonyos esetekben ugyanannak a forráskódnak különböző P-kód felel meg. A DECOMPILER utolsó verziója több mint 5000 Pascal forrássorból áll, és egy 100 kbájtos program (csak a pszeudokód) visszafordítása is kb. 1 óráig eltart — 286-os AT gépen. A viszonylagos lassúságot a Turbo Pascal kódszegmenseinek korlátozott nagysága miatti overlay-technika alkalmazása (és az 1.0-s verziószámból adódó, itt-ott redundáns kódolás) okozza.

A decompiler az egy tárgymodulba tartozó eljárásokat és függvényeket az object modul nevével azonos nevű, .PRG kiterjesztésű fájlba fordítja. Fordítás közben elkészül egy LINKER.BAT és COMP.BAT batch program. Ezek objectenként sorbarendezetten tartalmazzák a fordítási opciókat és az összeszerkesztési eljárást.

Például egy három OBJECT modulból álló program esetében:

```
COMP.BAT LINKER.BAT
Clipper obj1 -m -l LINK
obj1+obj2+obj3,PRGNEV,,Clipper
Clipper obj2 -m
Clipper obj3 -m -l
```

A DECOMPILER jelenlegi verziója -l opcióval fordított programoknál 95%-os, -l opció nélkül 98%-os helyességgel dolgozik. A -l (sorszám nélküli) fordítás esetén a nehézséget az optimalizált ciklusok visszaállítása okozza, mivel ugyanaz a bájtsorozat mást és mást jelenthet. Ennek a problémának a feloldása becslésem szerint hónapokba kerülhet, így jelen pillanatban ezt a maximum 5%-ot kézzel állítom helyre. („Művem" 2.0-s verziója ennek a problémának a feloldását is magába foglalja majd.)

A sorszámmal fordított programoknál a ciklusok visszaállítása nem okoz problémát, itt a compilerek különböző verzióinak veremkezelése jelenti a nehézséget.

Az ERR kiterjesztésű állományok memóriacímenként mutatják a generált forráskód elhelyezkedését, különösen figyelve a ciklusok mutatóira.

A jelenlegi fejlesztés a nem Clipper nyelvű modulok, C és ASSEMBLY szegmensek OBJECT formátumra hozását célozza meg. Ez feloldja az 1.0-s verzió korlátját, ami a nem STANDARD könyvtári (Clipper.LIB, EXTEND.LIB,PROCLIP2.LIB,TR.LIB ...) tárgymodulok újraszerkeszthetőségét teremti meg.

A CONTROL modul

Az ellenőrző fázis az eredeti és a visszafordított EXE/OVL programok bájtonkénti összehasonlítására való. A FULLMAP ellenőrző menetében készül el a .MAC kiterjesztésű kontroll MAP fájl, és a XXXXXCTR.XXX szegmensenkénti kontrollállomány. Az ellenőrző fázist a PROGRAM/ OVERLAY.CTR fájl vezérli.

Az összehasonlítási folyamat alatt észlelt hibák a XXXXXERR.DMP fájlokba íródnak be. Ez alapján állítható helyre a hibás program. Külön eljárás ellenőrzi a szimbólumtáblák azonosságát. Ezek különbözősége hívja fel a figyelmet az EXTERN hivatkozások jelenlétére.

A ciklusok helyességét egy LOOP_CHECK modul vizsgálja. Ez a compiler ciklusokkal kapcsolatos hibaelemzésével megegyező hibavizsgálatot végez, és a sorszám nélküli programszegmensek helyreállítását segíti.

A MESSAGE.ERO az egyes programszegmensek hibastátuszát tartalmazza.

A vírusok is ragadnak a mézre

A DECOMPILER kiválóan alkalmas vírussal fertőzött Clipper .EXE állományok mentesítésére, mivel a vírus az újraszerkesztett állományba nem kerül bele. Beszámolhatok olyan esetekről, amikor a forrásállomány elvesztése vagy sérülése miatt fordultak hozzám segítségért, s az eredmény megnyugtató volt.

A Clipper 5.0-s verziója sem oldotta meg azt, hogy csak azok a szükséges RUN-TIME rutinok szerkesztődjenek be az OUTPUT fájlba, amelyekre az adott programnak szüksége van, mert a COMMENT program az 5.0-s verzióval fordítva és szerkesztve majdnem 110 kbájt méretű lett (elhagyva minden fordítási és szerkesztési opciót).

A témával kapcsolatban az érdeklődőknek szívesen áll rendelkezésére:

Báró Csaba

Megtagadott alma mater

Az Alaplap 1990. szeptemberi számában megjelent Elaggott alma mater című cikkre az „aquila non captat muscas" ókori bölcsesség alapján először nem akartunk reagálni. Kiderült azonban, hogy vannak „sasok", akik még az ilyen szemtelen és apró legyet is „megfogják", mint az említett cikk szerzője. Történt ugyanis, hogy a szakmát oktató tanárok országos továbbképzésén az előadó, a cikk tartalmára hivatkozva, olyan szintű előadást tartott, amilyet ugyanezek a tanárok az első néhány órán tartanak a 14 éves elsősöknek, mondván: néhány alapfogalmat tisztázni kell.

Nem kívánunk itt annak az elemzésébe merülni, hogy egy volt diák, akinek az érettségi bizonyítványán még meg sem száradt a tinta, milyen alapon hurcolja meg iskolája és tanárai nevét. Elegendő ok-e erre, hogy mint minden diáknak, neki is volt kedvenc és kevésbé szívlelt tanára. Az sem feladatunk, hogy eldöntsük: egy szakmai — és nem pletykaorientált — lap tekintélyének használ-e, hogy ellenőrizetlen információkat közöl.

Inkább néhány — valóban bizonyítható — tény közlésére szorítkozunk, amit akár helyesbítésnek is lehet tekinteni.

Iskolánkban két évtizede, az országban elsőként indult meg a számítástechnikai szakemberképzés középfokon. Akkor valóban egyetlen R 20-as géppel indultunk, azonban az akkori lehetőségekhez képest ez nem számított sem korszerűtlennek, sem kis teljesítményűnek.

Ezután természetesen bennünket is elért a HT, Commodore s egyéb divathullám, de az iskolának inkább dicséretére válik, mint szégyenére, hogy nem váltogatta évenként a gépeket, operációs rendszereket, programnyelveket. Tanulóink hozzáférhettek egy-egy ilyen masinához pl. a szakkörökön, de a tananyagot alapvetően nem változtattuk. Ma is meggyőződésünk, hogy a középfokú szakemberképzés nem lehet olyan erősen gépfüggő, hogy egy életre vagy hosszú időre determinálná a programozó további működését az, hogy milyen gépen kezdte a tanulást.

Más dolog az, hogy a ma erősen szétváló kisgépes és nagygépes szemléletet egyformán meg kell ismertetni a tanulókkal. Erre ma a következő eszközbázis áll rendelkezésünkre:

A két nagyszámítógép (R 46 és IBM 4341) egyenként 8 megabájtos tárkapacitásával és a hozzájuk kapcsolt 4 db 100 megabájtos lemezegység, valamint egy 560 megabájtos winchester megfelelő háttér a programozást tanuló diákok számára.

A gépekhez való hozzáférést kétféleképpen oldottuk meg. Egyrészt kötegelt feldolgozás esetén a floppylemezeken szövegfájlban rögzített programokat egy átalakító berendezés segítségével csoportosan futtatjuk a gépen. Másrészt, ezzel akár egy időben a másik gép a jelenlegi 14 terminálon keresztül (bővítés 32-re a következő tanév kezdetére) interaktív kapcsolatban áll a programozásórán résztvevő tanulókkal.

A fentiekkel párhuzamosan oktatási terveink alapján IBM PC-ken (26 db) DOS és felhasználói programok oktatása folyik.

Visszatérve az elaggott alma materhez, néhány további tényről még érdemes említést tenni. Ha csak arról beszélnénk, hogy hány ma elismert szakember kezdte nálunk a szakmával való ismerkedését, akkor az olyan ellenőrizhetetlen zsurnaliszta fogásnak tűnne, mint amilyennel az előző cikk operál. Inkább a közelmúlt tényei:

— Amióta az Országos Szakközépiskolai Tanulmányi Verseny létezik e tárgyból, mindössze egy év volt, amikor az első helyezést nem a mi tanulóink érték el. (1989-ben például az első négy az iskola egyetlen osztályából volt)

— Szakirányú továbbtanulás esetén csak 1-2 tanulónak nem sikerül azonnal bekerülnie az egyetemre vagy főiskolára a jelentkező 8-10-ből. S akik nem tanulnak tovább, zömmel a szakmában helyezkednek el. Kivételt csak azok képeznek, akikről már az elején láttuk, hogy pályatévesztettek, de szerencsétlen oktatáspolitikánk nagyon megnehezíti az idejekorán történő pályamódosítást, s így aztán az érettségi kedvéért maradnak. Eddigi ismereteink szerint azonban ezek közül sem lett mindenki múzeumőr. Legfeljebb, akinek nem megy a programozás, az megpróbálkozik az újságírással.

Dr. Szám Lászlóné
tanár,
programozás munkaközösség vezető
Hámán Kató Közgazdasági
Szakközépiskola
1027 Budapest, II., Jurányi u. 1.

Tisztelt Tanárnő!

Kérésének megfelelően — kvázi helyesbítésként — közzétesszük az Ön levelét is, mint ahogy az egykori nebuló, Vámos Sanyi szubjektív hangvételű — de jól megírt — anyagának is teret adtunk annak idején. A tényfeltáráshoz hozzátartozik a vélemények feltárása is, azok pedig gyakran nagyon is eltérőek lehetnek, attól függően, hogy ugyanazokat a tényeket milyen megközelítésben veszik szemügyre. Érthető tehát a nézőpontok eltérése tanár és diák, vagy más-más nemzedékek között!

Véleményemet — szintén szubjektív módon! — egy mondatban szeretném összefoglalni: ha annak idején boldogult tanár koromban bárki a diákok közül kritikával illette — akár áttételesen is — munkámat, s ennek parányi valóságmagvát is felfedezni véltem, mindig változtattam eredetileg megfellebbezhetetlennek vélt álláspontomon. Igaz, nem is maradtam tanár, lett belőlem újságíró, s lehetek talán még múzeumőr is. De nem mindegy, hogy milyen múzeumban!

Üdvözlettel, sértődés nélkül:

Varga János

Dátum — pontosítva

Februári számunk 56. oldalán jelent meg a Dátumok előállítása képlettel című cikk. Sajnos a többszörös szövegkonverzió olykor már követhetetlen útvesztőiben az első képletből néhány karakter az enyészeté lett. Most igyekszünk a szerző által hibátlanul elkészített programsorokat mi is hiánytalanul közreadni:

```
Julnap = INT(365.25*UJEV) + INT(30.6001*UJHONAP + NAP + 1720982

ahol

UJEV = {  EV - 1   ha a HONAP = 1 vagy 2
          EV       ha a HONAP > 2
UJHONAP = {  HONAP + 13   ha a HONAP = 1 vagy 2
             HONAP + 1    ha a HONAP > 2
```

Dilis a Diri?

Az 1991. februári Alaplapban megjelent cikknek és reklámnak a hatására 1991. február 22-én vettem a magyar SolarSoft programok közül 2 darabot — készpénzért.

A shareware programok az Önök leírt reklámja szerint jól használható szoftvertermékek. Megjegyzem, én is így gondoltam, ismerve más, Önök által forgalmazott, nem magyar szerzőtől származó termékeket.

Ezzel szemben az egyik program, a Diri nevű, a következő apróbb vicceket engedi meg magának:

— Bejelentkezéskor erőszakkal átírja a rendszerdátumot 1990. február 22-re, s erről nem lehet lebeszélni.

— Pár perces munka után bejelenti, hogy „Sajnálom, ez demo példány! Emiatt Ön mennyiségi vagy egyéb korlátba ütközött! Vásárolja meg a rendszert!"

— Kilépve a generálszervizbe és nemleges válasz után kilépve a DOS-ba, hagy egy kedves üzenetet: FROS JOLLY line0, RUN error. Ezzel minden adat elvész, de hát nagy dolog, mert ennél még szebb ajándék is található a programban.

— Hiba nélküli kilépés esetén valami „ottfelejtődik" a RAM-ban, csekély 5792 bájt hosszon és természetesen Unknown néven, s egyben ráül az INT 05, 13, 14, 15, 17, 1C, 28 és 2F vektorokra. Megjegyzem, én ilyet még nem láttam, biztosan figyelmetlen voltam eddig, de hát így is lehet tanulni!

Én azt hittem, hogy az Önök pénztárába befizetett összeg vásárlás jellegű volt, de ki tudja?

A másik program, a BLISS, már nem felejt maga után semmit a RAM-ban, de üzenetet ő is küld, mintegy háromperces munka után, bár már előtte közölte, hogy valamit meg lehet vásárolni. Ez az üzenet nem más, mint:

Súlyos hiba a rendszerben, értesítse a programozót!!!

Jegyezze fel a következőket: 111 fk_110.c 472

Szerintem nagyon aranyos, nem beszélve valamilyen hivatkozott dokumentációról, amely sehol sincs, holott én legálisan vettem a programot. Legalábbis úgy gondoltam. Az is érdekes, hogy nem minden menüpont működik. Viccelődik a szerző, hogy ő milyen kiváló programozó, meg a programnak adott része milyen zseniális, csak éppen nem kívánja senkinek megmutatni.

Egyúttal büntetőjogi felelősségem tudatában bejelentem, hogy a megvásárolt programokat az első gépbe helyezés előtt írásvédetté tettem. A programok által okozott esetleges kárért, miután még nem derült ki, hogy az a bizonyos név nélküli, irattárban maradó program milyen kárt okozott, minden felelősséget Önökre hárítok.

Zala Péter

Bár Zala Péternek a levelet illetékes kollégánk már megválaszolta, annak közzététele — úgy érezzük — nincs minden tanulság nélkül. Egy félrevezető szerkesztési hibáért is szeretnénk elnézést kérni az esetleg bosszús felhasználóktól. A DIRI-ről szóló cikk ugyanis a shareware-szekcióval foglalkozó Közkincs rovatban kapott helyet, de az írás a teljes értékű kereskedelmi verzióra koncentrált s nem a shareware-változatra. Így érthető Zala Péter felháborodása, aki azt hitte, teljes értékű szoftvert vásárolt — shareware-áron.

Viszont a SolarSoft magyar shareware-szekciójának ez a — szerintünk zseniális — darabja valóban jelentős korlátozásokat tartalmaz, s ezzel eleget tesz a shareware koncepció szerinti szoftverforgalmazásnak. Tény persze, hogy csak az jár igazán jól, aki a későbbiekben meg is veszi a korlátozásoktól mentes teljes programot, hiszen annak, aki felmutatja a SolarSoft lemezt, a 28 000 Ft-os vételár helyett csak 20 000 forintot kell fizetnie.

A program magában foglal egy — C-ben írott — real-time órarutint is. Amennyiben nem a szabályos kilépési ponton hagyjuk el a programot, úgy ez az apró programrészlet is bennreked a memóriában. Mindenki megnyugodhat, teljesen ártalmatlan. Eltávolítása azonban csak a gép újraindításával lehetséges.

A DIRI legfrissebb, 5.0-s shareware változatával is megtisztelte a SolarSoft programkönyvtárt a DIRI programozója. Kibővített angol és német szótárral is felvértezte szoftverét.

A BLISS program szerzőjét értesítettük az Ön által tapasztalt kellemetlenségről, reméljük, mire ezeket a sorokat olvassa, a programfejlesztője az észlelt hiba okát már rég megszüntette.

Herczeg József

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk. A díjszabás kereskedelmi tevékenységet folytatóknak gépelt soronként (60 karakter) 100 Ft, másoknak az első sor 50,- Ft, minden további sor 20,- Ft.

Kérjük, hogy a hirdetés díját a Budapest Bank Rt-nél vezetett 380-66760 sz. Cédrus Rt. számlára utalják át, vagy postautalványon a Cédrus Rt. címére (1251 Budapest I., Lánchíd u. 15-17.) fizessék be, a hátoldalán feltüntetve, hogy apróhirdetés. A befizetést igazoló szelvényt a közlendő hirdetési szöveggel együtt az Alaplap szerkesztőségéhez küldjék el: 1251 Budapest XIV., Erzsébet királyné útja 17.

ADOK

Amigára eladó több mint 2000 lemez, játék- és felhasználói program. 5,25" -3,5" lemezek 380 és 750 Ft-os áron eladók. Keresztes Gábor, 1142 Budapest, Laky-köz 11., Tel.: 251-2523

Amiga 500 1 mb-ra bővítve eladó. Irányár: 53 900 Ft. Keresztes Gábor, 1142 Budapest, Laky-köz 11. Tel.: 251-2523

Amiga programok (30 Ft/lemez), 3,5"-es lemezek (75 Ft/db) eladók. Válaszborítékért listát küldök. Új Amiga 500-as számítógép (52 000 Ft), 512 kbájtos órás bővítő (7500 Ft), 3,5"-es floppy meghajtó (13 000 Ft) eladók. PPK, 7632 Pécs, Bókai J. u. 32.

AMI angol-magyar interaktív számítógépes szótár 37 000 címszóval és mintegy 23 000 angol kifejezéssel IBM PC XT/AT-re megrendelhető. Értékesítésre ügynököt, menedzsert keresek. Tel.: 133-2429; 115-4352

Eladó C64-hez ACTION REPLAY MK5, MK6, MK7 és ATOMIC POWER törőkártyák, valamint AMIGÁHOZ memóriabővítő. Tel.: 132-7473

Hello C64-players! Óriási lehetőség játékszerzésre! Felszámolás miatt eladó 400 lemezből álló játékpark. Csupa 1989-90-91-es játék! Gyorsan írj árajánlatot, nehogy más megelőzzön! 400 lemez + 500 játék! Garantied High Quality! Jelige: No lame game! A cím, ahol minden álmod valóra válik: Halmai Balázs, 9400 Sopron, Széchenyi tér 15.

Elromlott C64-es tápegysége? Féláron megjavítom! Csak 1250 Ft + postai utánvét! Küldje el postán, és én 10 napon belül visszaküldöm megjavított tápegységét! 1 év garancia! Az országban csak nálam! Izbéki Tibor, 5452 Mesterszállás, Ady Endre út 9.

Programok rendkívül kedvező áron C64, Plus/4, Enterprise, IBM gépekre. Cím: Varjas Zoltán, 2030 Érd, Emőke u. 1/b., I. em. 6.

ENTERPRISE-osok figyelem! Több mint 1000 program rendkívül olcsón és megbízható minőségben eladó! 3,5"-es (720 k) és 5,25"-es (720 k) lemezekre is! Széles választék a régebbi és a legújabb felhasználói, illetve játékprogramok között. Listát küldök (válaszboríték ellenében, de személyesen is), csere esetén kérek. Cím: Tóth Gusztáv, 1156 Budapest, Nádastópark 32.

ENTERPRISE programok eladók. Válaszborítékért listát küldök. 2000 program, sok kedvezmény, ajándékok. Cím: Zemen László, 1104 Budapest, Kada u. 141., fszt. 9.

Eladó: ENTERPRISE 128 + magnó + 720 k-os, 3,5"-es gyári floppy + monochrom monitor + szakirodalom (1800 Ft) + 10 db 3 M-ás floppylemez programokkal + programkazetták. Ár: 35 000 Ft. Érdeklődni levélben lehet: Fazekas Zoltán, 9026 Győr, Kilián út 12.

PRIMO játékprogramok (10 Ft/db) eladók. Balla János, 2459 Rácalmás, Népfront u. 19/1.

Profi sakkadatbank egy profitól: 40 000 keresztjátszma (!!), valamint az összes Karpov-Kaszparov meccs is. Okosabb, mint egy sakkmester. Spectrumon (Speccy Dos is), majd ENTERPRISE-on, IBM-en tervezve. Atari ST, Amiga stb. ifj. Lovass László, tel.: 183-6516

Jodál Endre:
Számítástechnikai kislexikon
(Bukarest, 1990. Kriterion Könyvkiadó, 469 oldal. Ára: 45,— Lej)

Csak néhány napra kaptam kölcsön a Kriterion Kézikönyvek sorában megjelent Számítástechnikai Kislexikont, melynek létezéséről egy, az Élet és Tudományban közölt recenzióból szereztem tudomást, és amelyért hiába jártam végig nagyobb budapesti könyvesboltokat. Módszeres végigolvasására nem volt lehetőségem, csupán tallóztam a címszavak között.

A kislexikon a számítástechnika sok területét felöleli, bár teljességről terjedelmi okokból sem lehet szó. A címszavak után zárójelben van a megfelelő román és angol fordítás. A „visszafelé keresést" a kötet végén található román—magyar és angol—magyar szótár segíti. Az egyes címszavakhoz tartozó magyarázatok pontosak és rendkívül tömörek. A túlzott tömörség olykor nehezítheti a megértést. (Egy példa: a számfüggvény leírásánál a témát nem ismerő hibának véli, hogy a második elemhez az 1 érték tartozik, mert nem tudja, hogy a számozás nullával kezdődik.) A tömörség zavaró hatását némiképp ellensúlyozza a számos szembetűnő kereszthivatkozás, azaz más címszavakra való utalás.

Néhány meghonosodott angol szó (például fájl, kurzor) csak a kötet végi szótár segítségével található meg, az utóbbi a helyőr címszó alatt. Találkoztam néhány — legalábbis számomra — eddig ismeretlen fogalommal is, ezek közül nagyon tetszett a szemléletes „forró krumpli" adatútképzés.

Nagy hibája ennek a könyvnek az alacsony megjelenési példányszám. Az impresszumban jelzett 4070 példányból nemcsak a magyarországi könyvesboltokba nem jutott, de már Romániában sem kapható. (Összehasonlításul: az ugyanebben a sorozatban 1973-ban kiadott Mediterrán mítoszok és mondák 27 255 példányban jelent meg.) Remélem, hogy egy újabb kiadás révén a magyarországi érdeklődők is hozzájuthatnak az érdekes és hasznos kötethez.

(bl)

Peter Norton:
DOS-kalauz
(Budapest, 1990. Novotrade Kiadó — Brady Books, 326 oldal. Ára: 469,— Ft)

A minden eddiginél jobb DOS-kalauz.

Ismerje meg a DOS-t Peter Norton irányításával!

Ez a kiadvány, amely Peter Norton eddig legsikeresebb, már-már klasszikus kötetének teljesen átdolgozott és naprakész változata, tele van hasznos és gyakorlati tanácsokkal.

Hosszú évek tapasztalatainak birtokában Peter Norton úgy közelít a számítógépekhez, hogy az mindenki számára érthető legyen. A könyvből minden lényeges tudnivaló megtanulható ahhoz, hogy számítógépünk operációs rendszerét a lehető leghatékonyabban működtessük. Ehhez nyújtanak további segítséget Peter Norton saját ötletei és módszerei, amelyeknek a szerző méltán megérdemelt hírnevét köszönheti.

A DOS megismerésének legegyszerűbb és egyben legélvezetesebb módja, ha az utat magával a nagymesterrel, Peter Nortonnal járja végig.

E havi összeállításunkban a hónap témájához kapcsolódóan számítástechnikai szótárak, értelmező szótárak és lexikonok között válogattunk.

Az IFIP-ICC Információfeldolgozási értelmező szótára. Budapest, 1968. Országos Ügyvitelgépesítési Felügyelet, 294 oldal. Ára: 45,— Ft

Frey Tamás — Szelezsán János: Számítástechnika (Műszaki Értelmező Szótár 33.). Budapest, 1973. Akadémiai Kiadó, 168 oldal. Ára: 45,— Ft

Frey Tamás — Szelezsán János: Matematikai Kibernetika (Műszaki Értelmező Szótár 34.). Budapest, 1973. Akadémiai Kiadó, 120 oldal. Ára: 33,— Ft

Löbel, P. — Müller, G. — Schmid: Számítástechnikai kislexikon. Budapest, 1973. Műszaki Könyvkiadó, 465 + 20 oldal. Ára: 89,— Ft

Adatfeldolgozás, Programozás (Datenerfassung, Programmierung) Technik Wörterbuch (Englisch, Deutsch, Franzözisch, Russisch, Ungarisch). Berlin, 1976. VEB Technik, 668 oldal. Ára: 200,— Ft

Dr. Ádám Sándor: Népszerű elektronikai minilexikon. Budapest, 1985. LSI ATSZ, 211 oldal. Ára: 198,— Ft

Kis Ádám (szerk.): Mi micsoda magyarul a számítástechnikában? Számítógépes értelmező szótár. Budapest, 1986. Tömegkommunikációs Kutatóközpont, 171 oldal. Ára 71,— Ft

Dr. Ádám Sándor: Népszerű számítástechnikai minilexikon. Budapest, 1988. Magánkiadás, 252 oldal. Ára: 169,— Ft

Dr. Kovács Magda: Mikroszámítógép-mikroelektronikai szótár I. Angol—magyar szakszótár. Budapest, 1988. LSI ATSZ, 250 oldal. Ára: 197,— Ft

Dr. Kovács Magda: Mikroszámítógép-mikroelektronikai szótár III/1—3. English — Deutsch — Espanol — Esperanto — Magyar. Budapest, 1989. LSI ATSZ, 840 oldal. Ára: 820,— Ft

Dr. Kovács Magda: Mikroszámítógép-mikroelektronikai szótár VI. English — Deutsch — Magyar. Budapest, 1990. LSI OMAK Alapítvány, 381 oldal. Ára: 489,— Ft

Oxford számítástechnikai értelmező szótár. Budapest, 1989. Novotrade Kiadó, 510 oldal. Ára: 490,— Ft

Holland, R. C.: Mikroelektronika és mikroszámítógépek. Illusztrált értelmező szótár. Budapest, 1990. Műszaki Könyvkiadó, 152 oldal. Ára: 195,— Ft

Homonnay Péter: Angol—magyar számítástechnikai szótár (3. kiadás). Budapest, 1990. Novotrade Rt, 287 oldal. Ára: 79,— Ft

Jodál Endre: Számítástechnikai kislexikon. Bukarest, 1990. Kriterion Könyvkiadó, 469 oldal. Ára: 45,— Lej

INFORMÁCIÓKÉRÉS: 56 ▲



Vitamindús tavaszi menü

Amikor a tavasz közeleg, valahogyan minden más arcot ölt. Mintha még a kormos városok is megirigyelnék a természet színeváltozását. Mintha az egész telet csak arra találták volna ki, hogy legyen mihez éles kontrasztot állítani. A műhelyekben folyó, visszavonult, viszonylag csendes munka után a számítástechnikai szakma is nyújtózott egy jókorát. Jól példázta ezt az az eseménydömping, amely a miskolci MikroCAD február végi–március eleji időpontján csúcsosodott ki: a szakma szinte két táborra szakadt — az egyik ott volt Miskolcon, a másik pedig, mivel nem volt ott, mindenáron hallatni kívánt magáról valami más formában. Szinte egymást érték a sajtótájékoztatók, bemutatók.

Multiplus(ssz)

Ha valaki kinövi amúgy még jó állapotban lévő télikabátját, s nincs kinek ajándékoznia — meg nem is engedheti meg magának —, fogja és elviszi a „bizóba". Ott néhány hónap múlva esedékes pár forintot bizonyára kap érte. De mi legyen, ha a winchesterünk kapacitását nőttük ki? Például egy használt, 40 megabájtos winchesterért legfeljebb 10—15 ezer forintot lehet kapni, az új, 120 megás pedig belekerül legalább 85 000-be.

A „ne dobd el a régit" jegyében a tárkapacitás megnövelésének egy lehetséges útját kínálja a Multicad Stúdió. Expanz! fantázianevű adattömörítő kártyájuk 32 500 forintért nyújtja — tehát közel a másik megoldás feléért — a 80 megabájtos ugrást. A kaliforniai InfoChip Inc. által gyártott kártya révén, az adattömörítési technológiának köszönhetően, a merevlemezes tárolók kapacitása akár háromszorosára, a hajlékonylemezes tárolóké pedig akár 15-szörösére(!) is növelhető. Előnye a megoldásnak, hogy a nagy kapacitású diszkeket is könnyedén kezeli, s hogy a „valós idejű" üzem miatt minden funkció automatikus, az adattömörítés és visszaalakítás folyamatosan, várakozás nélkül történik. A kártyához tartozik egy olyan segédprogram is, amely lehetővé teszi az Expanz! kikapcsolását, így a felhasználónak mindenkor lehetősége van arra, hogy hozzáférhessen DOS-állományokhoz, -partíciókhoz.

MULTICAD studio

Ha egy másik gépre kívánjuk átvinni az adatokat, és az nincs ellátva Expanz! kártyával, akkor sincs gond. A kártya alapbeállításának értelmében ugyanis eredeti — tehát nem tömörített — formában menti mágneslemezre az állományokat, s azok „csont nélkül" átvihetők egy másik PC-re. Ha viszont az információt fogadó gép is fel van szerelve a kártyával, egyetlen parancsot kell kiadni, hogy tömörített formában mentődjék lemezre az információ.

Egyetlen hátránya az Expanz!-nak — s erre a magyarországi forgalmazó Multicad külön fel is hívja a figyelmet —, hogy az MS-DOS-tól eltérő operációs rendszer (például Compaq-DOS) esetén előfordulhatnak együttműködési hibák.

Startolnak az óvodások

Van ára is annak az egylemeznyi szoftvernek — 2500 forint —, amely a miskolci MikroCAD-en az egyik első díjat nyerte el a vásárdíj-pályázaton, de a fejlesztő inkább azt szeretné, ha valamely nagyobb értékű termék társaságában, mintegy apró figyelmességként kerülhetne oda a legtöbb felhasználó programkönyvtárába. A program célközönsége ugyanis a család, azon belül is a 4—8 éves korosztálynak az a rétege, amelyik játékos úton, nem csupán játék értékű feladatok megoldására is fogékony.

A Start ugyanis — mint olvasható a „kézifüzet" címlapján — programozási környezet kisgyermekek számára, 4 éves kortól. A program használatához persze elengedhetetlen a szülői segítség, de csak az indulásnál! A gyereknek pedig olvasni sem kell tudnia, csak a számokat kell ismernie — ami joggal feltételezhető is — 1-től 10-ig. A programkörnyezet az ismerkedést követően úgyszólván kínálja magát: ismerj meg, próbálj ki, kísérd nyomon, hogyan is születik egy program. A Start használata során a gyerekek megismerhetik a számítógépnek adható elemi utasítások értelmét, az elemi utasításokból felépíthető programsor lényegét, az új elemek létrehozásához vezető programépítés logikáját, az elemek összeépíthetőségét és az abból adódó konstrukciós egységek feladatleegyszerűsítő hatását.

A Start program a ConTech Kft. alkotói műhelyében született, mi az egyik lehetséges forgalmazó, a Systrend Kft. standján láttuk Miskolcon.



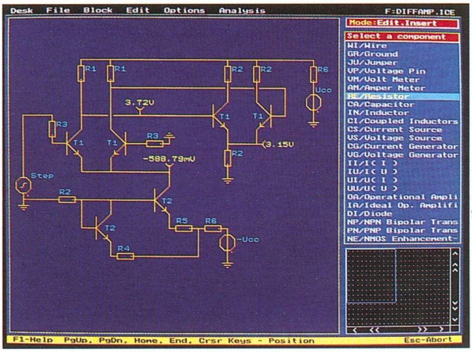
Tervszerű áramkörök

Ugyanezen a helyen még egy program megismerésére is lehetőségünk nyílott. A Tina nevű menüvezérelt, grafikus elektronikai tervező- és oktatóprogram igazi vargabetűvel jutott el a hazai piacra: magyar programozók készítették, Hollandiában majd Németországban volt a piaci premier, s csak az ottani sikereket követően került sor a magyar agyakban megfogant gondolatok itthoni adaptációjára is: mostanra már — mintegy 50 000 — kemény magyar forintért is megvásárolhatjuk. Forgalmazója a Rair Kft.

A Tina nem kifejezetten olcsó program, de célirányos alkalmazási területének, sokrétű szolgáltatásainak révén nagy népszerűségre tehet szert itthon is. Segítségével tetszőleges analóg áramkör felrajzolása és analízise lehetséges, az előzetes analízis jelentősen csökkenti a „deszkamodellek" megépítéséből és beindításából származó fejlesztési költségeket.

A Tina lehetővé teszi többek között az áramkörök válaszának (feszültség, áram, teljesítmény) tetszőleges áramköri paraméter, illetve hőmérséklet függvényében való ábrázolását, valamint áramköri paraméterek adott célfüggvény melletti automatikus meghatározását. A programhoz nagyméretű félvezető-katalógus tartozik, amelyet a felhasználó tovább bővíthet.

A program hatékonyan támogatja a különböző oktatási anyagok kidolgozását, illetve adott tananyag önálló elsajátítását. Az egyfelhasználós verzió mellett Novell Netware hálózati verzió, valamint olcsó (8000 Ft!) diák/hobbi verzió áll a felhasználók rendelkezésére. A fejlesztők dolgoznak azon, hogy kibővítsék a Tina külső illeszkedését, és a széles körben elterjedt ORCAD program fogadni tudja a Tina információit.



INFORMÁCIÓKÉRÉS: 45 ▼

Minőség — a Távol-Keletről is !

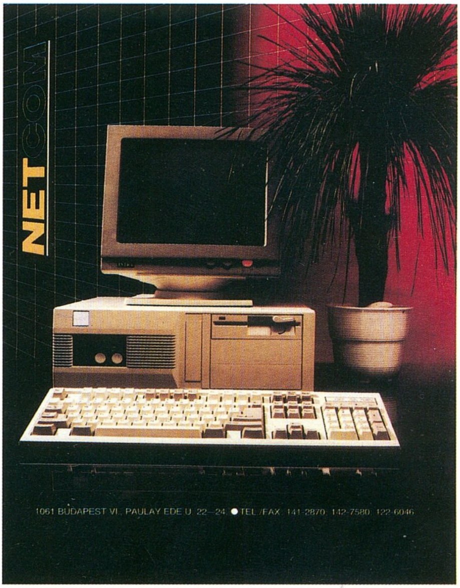
Március elején tartotta meg termékbemutatóját a NETCOM Számítástechnikai és Kereskedelmi Kft. Felvonultatták kínálatban mindazokat a hardverkategóriákat, amelyek termékeit ma Magyarországon a legtöbb, hardverértékesítéssel foglalkozó cég forgalmazza. Ezek a berendezések elsősorban a kisméretű irodák komplett felszerelésének lehetőségét kínálják. A NETCOM — amint ezt a bemutató is jelezte — elsősorban nem is a kínálat falrengető újdonságával, sokkal inkább üzletpolitikájának eredeti megfogalmazásával kíván kiemelkedni a hardvergyártók sorából.

Amióta egyáltalán létezik, a magyarországi hardverpiacon kétféle álláspont uralkodik. Egyes cégek az azonnali meggazdagodás reményében, dömpingszerűen ontják a „vasat", de még így is meglehetősen magas áron. Mások viszont hosszú távon gondolkodnak, ami azt jelenti, hogy az árfekvés a közepeshez közelít, a termékek magas műszaki színvonalúak, és mindenféle plusz szolgáltatással igyekeznek a vevő kedvében járni, elérni azt, hogy a vevő elégedett legyen velük, és újbóli vásárlás esetén ismét őket válassza. Mindkét koncepció híveinek kiélezett harcot kell vívniuk mind a hasonló, mind az eltérő elveket valló konkurens cégekkel.

A NETCOM-ot üzleti filozófiája a második táborhoz közelíti, fő erősségüknek azonban inkább azt vélik, hogy ez a filozófia igyekszik markánsan elkülönülni a többiekétől — ha úgy tetszik —, antropomorf jelleget ölteni. Náluk minden tevékenység személyhez kötődik. A vásárlás tanácsadással kezdődik, és a vevőnél ér véget, a gép bekapcsolásával. Névhez kötődő üzletek vannak, vagyis az a személy, aki megköti az üzletet, végigkíséri az adott termék garanciális életét, tehát a vevő mindig ugyanazzal a személlyel áll kapcsolatban. Ez egyrészt a vevő szempontjából rendkívül jó dolog, másrészt a cég szempontjából is módfelett előnyös, mert nyomon tudják követni, hogy melyik munkatársuk miként dolgozik. Többek között ennek az üzletpolitikának az eredménye, hogy az 1990 májusában alakult cég már tavaly 210 millió forintos forgalmat bonyolított le, 1991 első két hónapjában pedig újabb 85 millió járult a tavalyi eredményhez.

A bemutatott gépek Tajvanból és Szingapúrból származnak. A távol-keleti eredetű gépekről pedig ugye már mindenféle rosszat elmondtak. Ezek az olykor teljesen helytálló kritikák azonban egy nagyon fontos dologról megfeledkeztek: nevezetesen arról, hogy egy kalap alá venni „a" Távol-Keletet, az ottani gyártókat épp olyan hiba, mint ha valaki felületesen elfelejt különbséget tenni bit és bájt között. A NETCOM forgalmazta berendezések tehát igenis Tajvanról és Szingapúrból származnak, de többlépcsős minőségellenőrzésen mennek keresztül. A NETCOM szakemberei évente többször is a helyszínen, a gyári futószalagról választanak ki gépeket, és azokat szúrópróbaszerűen ellenőrzik. A gyári minőség-ellenőrzésen átment gépeket még egy, állandóan ott tartózkodó szakemberük is teszteli, s csak az ő pozitív véleménye után szállítják Magyarországra, ahol a vevőhöz történő kiszállítás előtt még egy utolsó tesztelésre kerül sor.

A termékbemutatón sokak érdeklődését felkeltette a HAWK munkaállomás. Ez a Highlight által fejlesztett „mindent egyre" kártya integrálja az alaplapot, a monokróm grafikus adaptert, 2 soros, 2 párhuzamos interfészt és egy ARCNET kártyát, mindezeket egy baby méretű lapon. Ebből az 59 900 Ft-os gépből rövid idő alatt 1200 darabot adtak el. Ezzel a rendkívül jó eredménnyel a Highlight Computer disztribútorai közötti világversenyen az előkelő 2. helyezést érték el.



Vevőcsalogató terméknek bizonyult az XT-LAPTOP, amely már 38 900 forintos áron kapható. Ez a home-computer kategóriájú gép 4,77/10 MHz órajelű, 640 k RAM, két 3 1/2 inches, 720 kbájtos floppyegység és egy modem a tartozéka. Első PC-kategóriájú gépként bizonyára sokan választják majd.

Egészen más tábort, az igényes, professzionális felhasználókat célozták meg a kiállított ALR EISA 386/33 géppel, azokat, akiknek a sebességi viszonyok, a folyamatos bővítés lehetősége nagyon fontos. A gép főbb jellemzőiről korábban a VEISA kapcsán, 1990/11-es számában már beszámolt az Alaplap. Ez a gép már valóban nem játékszer (elég csak a 8 megabájt RAM-ot, a 32 bites HDD kontrollert, a 330 megabájtos winchestert vagy a Mylex 32 bites Ethernet kártyát említeni).

A NETCOM lézerprintereket is forgalmaz, de ha már forgalmaz, akkor nem elégszik meg a puszta adásvétel lebonyolításával e kategória esetében sem. Hogy a termékhez kapcsolódó szolgáltatások körét bővítsék, megvásárolták Hongkongból a lézerprinterek festékszalagjainak felújítási technológiáját is. A megrendelő telefonhívására nála végzik el a szükséges cserét, illetve felújítást.

Perifériák széles skáláját is felvonultatták a termékbemutatón, s nem hiányoztak a termékek sorából a telefaxok sem. Újnak számítanak a választékban azok a számítógéppel összekötött faxok, amelyek rugalmasan ötvözik a két típus kínálta szolgáltatásokat. E speciális konfiguráció esetében a merevlemezre (merevlemezről) egy fájlba (fájlból) érkezik a szöveg, a számítógép billentyűzetéről pedig tárcsázni is lehet. Ha erre nincs szükség, akkor kikapcsolva közönséges faxként használható a berendezés.

A bemutatón részt vett még a Quadro Byte Kft. is, amely a NETCOM gépein mutatta be saját fejlesztésű szoftvertermékeit — mint például élelmezési, radiológiai, értékpapírnyilvántartó és főkönyvi rendszerét. Így tehát az érdeklődők munka közben is tesztelhették a gépeket. (x)



INFORMÁCIÓKÉRÉS: 41 ▲